

Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE0624 – Laboratorio de Microcontroladores
II ciclo 2024

Proyecto Final
Sistema de Parqueo Inteligente con IoT

Leonardo Serrano Arias C17484
Lorena Solís Extteny B97657

Profesor: Marco Villalta

27 de Noviembre, 2024

Índice

Introducción	3
Objetivos	3
Objetivos Generales	3
Objetivos Específicos	3
Alcances	3
Justificación	4
Marco Teórico	4
Internet de las Cosas (IoT)	4
ESP-WROOM-32	4
Registros y Periféricos	6
Servomotor SG90	7
Sensor de proximidad infrarrojo FC-51	8
I2C 1602 LCD	8
Librerías de Arduino IDE	9
Blynk	9
Lista de componentes	10
Diagrama de flujo del proyecto	10
Diagrama de bloques del proyecto	10
Desarrollo	11
Componentes y Funcionalidad	11
Funcionamiento del Sistema	12
Inicialización del Sistema	13
Gestión de Entrada de Vehículos	13
Gestión de Salida de Vehículos	13
Visualización en la Pantalla LCD	13
Control y Monitoreo Remoto con Blynk	14
Análisis de Resultados	15
Conclusiones y recomendaciones	15
Anexos	16

Índice de figuras

1. Diagrama de pines ESP32 [2]	5
2. Diagrama de bloques ESP32 [2]	6
3. Diagrama de potencia ESP32 [2]	6
4. Diagrama del Servomotor SG90 [3]	8
5. Diagrama del sensor de proximidad FC-51[6]	8
6. Diagrama de flujo del parqueo inteligente	10
7. Diagrama de bloques del parqueo inteligente	11
8. Maqueta del sistema de parqueo inteligente	12
9. Circuito en protoboard del sistema	12
10. Visualización del estado de disponibilidad de los espacios en la pantalla LCD . .	14

11. Visualización de espacios disponibles en la plataforma Blynk por conexión WiFi cuando el slot 2 está ocupado 14

Introducción

En el contexto actual, la congestión vehicular y la limitada disponibilidad de espacios de estacionamiento en áreas urbanas plantean un desafío significativo para la movilidad y la eficiencia en el uso del tiempo. Los sistemas de parqueo tradicionales suelen requerir supervisión manual, lo que genera demoras y poca transparencia en la gestión de espacios. Ante esta problemática, la implementación de sistemas de parqueo inteligentes se perfila como una solución innovadora y eficiente, aprovechando tecnologías como el Internet de las Cosas (IoT), sensores y plataformas digitales para optimizar el proceso de estacionamiento. Este proyecto busca desarrollar un sistema de parqueo inteligente que automatice la detección de vehículos, la gestión de espacios y la visualización de datos en tiempo real, mejorando la experiencia del usuario y optimizando los recursos disponibles.

El sistema propuesto utiliza un microcontrolador ESP32 como núcleo para coordinar múltiples sensores infrarrojos, servomotores y una pantalla LCD, integrándose con la plataforma móvil Blynk para la monitorización y reserva remota de espacios. Este enfoque permite a los usuarios visualizar la disponibilidad de slots de parqueo en tiempo real tanto en una pantalla local como en sus dispositivos móviles, además de controlar el acceso mediante un mecanismo automatizado de agujas en las entradas y salidas. Con la posibilidad de escalar este sistema y adaptarlo a diferentes entornos, el proyecto no solo contribuye a una mejor gestión de estacionamientos, sino que también sienta las bases para futuras aplicaciones de IoT en el ámbito urbano.

El repositorio utilizado para el desarrollo de este proyecto se encuentra en la siguiente dirección: https://github.com/Leonardo-SA/IE-0624_II_2024_C17484_B97657.git

Objetivos

Objetivos Generales

- Desarrollar un sistema de parqueo inteligente que permita consultar en tiempo real la disponibilidad de espacios, por medio de una aplicación IoT.

Objetivos Específicos

1. Implementar un sistema de monitoreo de un parqueo en tiempo real, mediante el uso de un microcontrolador ESP32 y la correcta implementación de un servomotor, así como distintos sensores infrarrojos.
2. Diseñar una interfaz de usuario para visualizar el estado del estacionamiento y reservar espacios en el mismo.
3. Integrar la conectividad WiFi para sincronizar el estado del parqueo con la nube y contar con un sistema de reservas.
4. Crear una maqueta del parqueo con agujas automáticas de entrada y salida.

Alcances

- Desarrollar un sistema basado en hardware económico que sea sencilla de implementar y configurar.
- Garantizar la comunicación entre los dispositivos y la aplicación IoT.

- Proporcionar una interfaz intuitiva donde se pueda visualizar los espacios del parqueo en un dispositivo con acceso a internet.

Justificación

Un sistema de parqueo inteligente tiene como objetivo solucionar problemas comunes en los estacionamientos tradicionales, como la falta de espacios, el uso inefficiente de los recursos y la pérdida de tiempo en la búsqueda de espacios disponibles. Este tipo de sistemas es especialmente relevante en entornos urbanos y zonas comerciales, donde gestionar eficientemente el espacio es fundamental para optimizar el control interno del parqueo, mejorar la experiencia del usuario y agilizar el proceso de estacionamiento.

En este proyecto se procurará desarrollar este sistema mediante el uso de un ESP32 como microcontrolador central, que ofrece conectividad Wi-Fi integrada, lo cual es clave para implementar el concepto de Internet de las Cosas (IoT). A través de sensores infrarrojos, el sistema podrá detectar la presencia o ausencia de vehículos en cada espacio de estacionamiento, lo cual se mostrará en una pantalla LCD. Además, un servomotor controlará la apertura y cierre de barreras automáticas para regular el acceso a los espacios.

El uso de la plataforma IoT permitirá que el sistema se conecte a una red y proporcione datos en tiempo real sobre la disponibilidad de espacios a los conductores, lo que optimiza el flujo vehicular y reduce la frustración de los usuarios al buscar un lugar para estacionar. Esta información podrá ser accesible desde dispositivos móviles, lo que facilita la toma de decisiones al usuario antes de llegar al estacionamiento.

Marco Teórico

Internet de las Cosas (IoT)

El término Internet de las cosas (IoT) hace referencia a la red de dispositivos conectados que intercambian datos entre sí y con la nube, gracias a tecnologías avanzadas y al desarrollo de chips de bajo costo. Este concepto permite que objetos cotidianos, como electrodomésticos, vehículos y herramientas, utilicen sensores para recopilar datos y responder de manera inteligente a los usuarios. [1]

Desde la década de 1990, los avances en miniaturización y eficiencia de los chips han impulsado la incorporación de sensores y procesadores en objetos comunes. Inicialmente, esta integración era limitada debido al tamaño y costo de los dispositivos electrónicos. Con el desarrollo de tecnologías como las etiquetas RFID y chips más pequeños y potentes, la computación se volvió más accesible, permitiendo conectar incluso dispositivos simples, como interruptores de luz, a servicios avanzados de voz. Este ecosistema de dispositivos interconectados, capaces de transmitir y recibir datos automáticamente, constituye el núcleo del Internet de las cosas, transformando hogares, oficinas y empresas con soluciones tecnológicas inteligentes.[1]

ESP-WROOM-32

El microcontrolador ESP-WROOM-32 es un módulo basado en la familia de chips ESP32 de Espressif System [2]. Este módulo es ideal para desarrollar proyectos de IoT debido a su precio accesible, eficiencia energética y avanzadas funcionalidades de conectividad inalámbrica.

Además, este cuenta con una tecnología de 40nm de TSMC, el ESP32 destaca por su rendimiento en términos de consumo energético como de radiofrecuencia (RF). El diseño del ESP32 incluye componentes como un conmutador de antena, balun RF, amplificador de

potencia, amplificador de bajo ruido (LNA), filtros y módulos de gestión de energía. Además, el ESP32 integra circuitos avanzados de calibración que corrigen imperfecciones externas y se adaptan a cambios ambientales, eliminando la necesidad de equipos especializados para pruebas de conectividad Wi-Fi durante la producción masiva.

En términos de conectividad, el ESP32 soporta Wi-Fi en los estándares 802.11 b/g/n, con velocidades de hasta 150 Mbps en la banda de 2.4 GHz, y ofrece características como monitoreo de señales de baliza, fragmentación y reensamblaje de paquetes, así como diversidad de antena. Estas funcionalidades, junto con su capacidad para operar en modos simultáneos (Station, SoftAP y promiscuo), lo convierten en una solución flexible y eficiente para proyectos IoT de última generación. Algunas características principales son:

- **Procesador:** Doble núcleo con arquitectura de 32 bits, basado en el procesador Tensilica Xtensa, con frecuencia de hasta 240 MHz
- **Memoria:** Cuenta con 520 KB de SRAM y hasta 16 MB de memoria flash.
- **Conectividad:** Soporta tanto Wi-Fi (802.11 b/g/n) como Bluetooth 4.2 (Classic y BLE).
- **I/O:** Dispone de múltiples GPIOs (pines de entrada/salida digital), entradas analógicas, y soporte para interfaces como SPI, I2C, UART, entre otras.
- **Consumo de energía:** Es eficiente en cuanto a consumo, con múltiples modos de ahorro de energía que permiten su uso en aplicaciones alimentadas por batería.

A continuación, se muestra la distribución de pines del microcontrolador ESP32. El cual cuenta con 49 pines, 36 pines configurables para I/O, alimentación, ADC/DAC, comunicación y control, como se muestra en la figura 1.

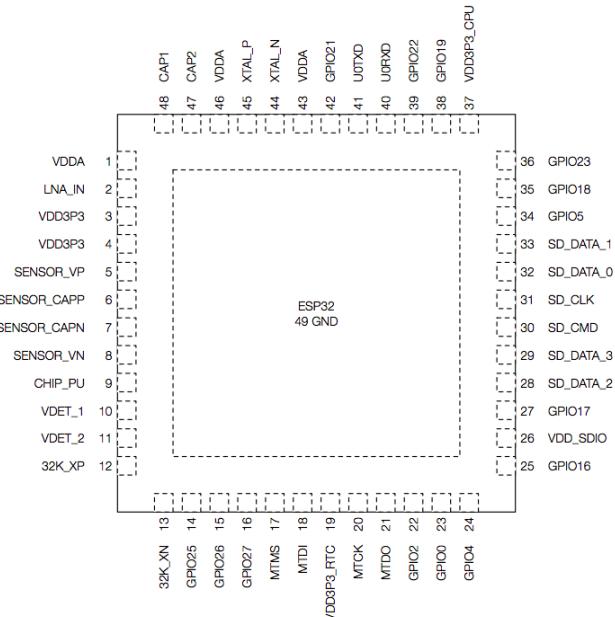


Figura 1: Diagrama de pines ESP32 [2]

Seguidamente, se observa en la figura 2 el diagrama de bloques del microcontrolador que muestra toda la arquitectura del mismo.

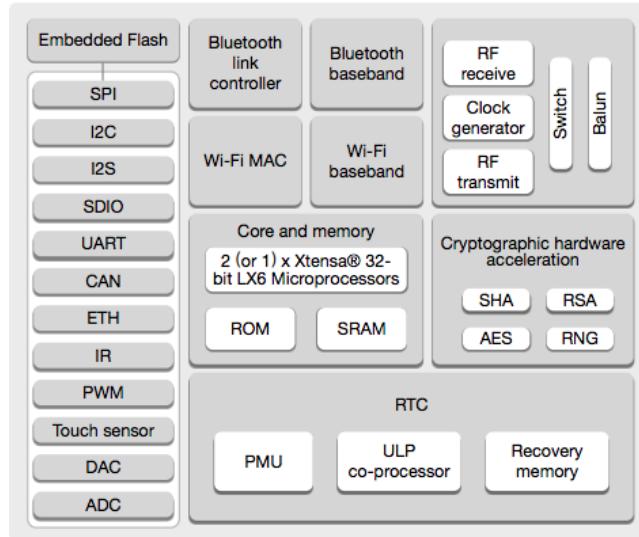


Figura 2: Diagrama de bloques ESP32 [2]

Finalmente, se tiene el esquema de potencia del ESP32 que cuenta con tres dominios: VDD3P3_RTC, VDD3P3_CPU y VDD_SDIO. VDD3P3_RTC alimenta el RTC y el CPU, mientras que VDD3P3_CPU suministra energía solo al CPU. VDD_SDIO está conectado a un LDO interno que puede configurarse a 1.8V o al mismo voltaje que VDD3P3_RTC, y se desactiva automáticamente si comparten la misma conexión. Además, el LDO puede apagarse por software en el modo Deep-sleep para reducir el consumo de energía.

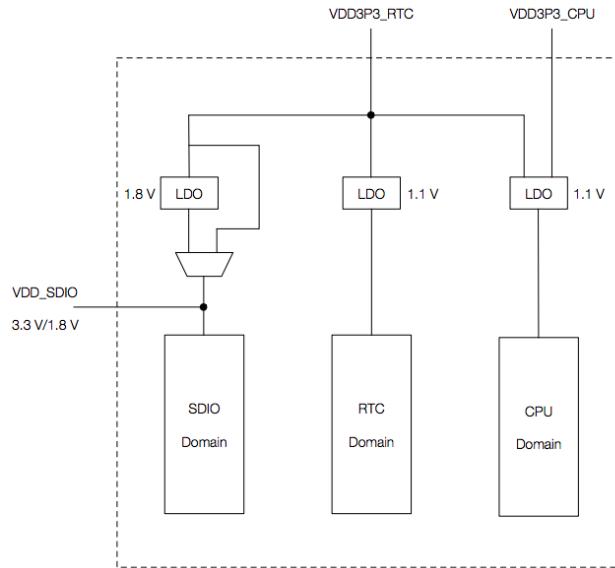


Figura 3: Diagrama de potencia ESP32 [2]

Registros y Periféricos

La tabla que se muestra presenta los registros y periféricos utilizados en el sistema de parqueo inteligente. Detalla los pines del microcontrolador ESP32, los dispositivos conectados, su función dentro del sistema y el modo en que operan (entrada, salida o comunicación).

Tablas 1: Registros y Periféricos del Sistema de Parqueo Inteligente

Pin/Registro	Periférico	Uso	Modo de Operación
25	Sensor digital infrarrojo	Detección de ocupación en el Slot 1	Entrada digital
26	Sensor digital infrarrojo	Detección de ocupación en el Slot 2	Entrada digital
27	Sensor digital infrarrojo	Detección de ocupación en el Slot 3	Entrada digital
32	Sensor digital infrarrojo	Detección de vehículos entrando al estacionamiento	Entrada digital
33	Sensor digital infrarrojo	Detección de vehículos saliendo del estacionamiento	Entrada digital
18	Servomotor	Control de barrera de entrada	Salida PWM
19	Servomotor	Control de barrera de salida	Salida PWM
34	Sensor digital infrarrojo	Señalización del estado de la barrera de entrada	Salida digital
35	Sensor digital infrarrojo	Señalización del estado de la barrera de salida	Salida digital
I2C (0x27)	Pantalla LCD	Visualización de la disponibilidad de espacios	Comunicación I2C
Wi-Fi	Módulo ESP32	Conexión a la plataforma Blynk	Comunicación inalámbrica

Servomotor SG90

El SG90 [3] es un pequeño servomotor ampliamente utilizado en proyectos de electrónica y robótica debido a su bajo costo y facilidad de control. Este servomotor tiene un rango de movimiento angular que típicamente va de 0° a 180°. Algunas de sus características principales son:

- **Torque:** Genera un torque de alrededor de 1.8 kg·cm a 5V.
- **Alimentación:** Funciona con una fuente de 4.8V a 5V.
- **Control:** El servomotor se controla mediante una señal de pulso de ancho variable (PWM) que determina el ángulo de rotación.
- **Velocidad:** 0.1s/60 degree

En figura 4 se muestra la funcionalidad de sus pines y un pequeño diagrama:

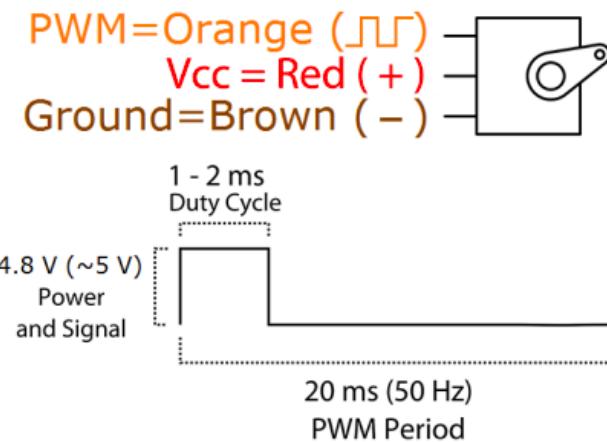


Figura 4: Diagrama del Servomotor SG90 [3]

Sensor de proximidad infrarrojo FC-51

Este módulo sensor infrarrojo emisor y receptor se puede adaptar a luz ambiente y distancia de detección a través de un potenciómetro que viene incluido en la placa. La distancia de detección está comprendida entre 2 cm y 30 cm, con un ángulo de detección de 35°. Estos infrarrojos emiten señales a cierta frecuencia cuando detectan algún obstáculo (superficie de reflexión). La señal captada por los sensores es acondicionada mediante un circuito comparador, lo que se refleja en un LED indicador de color verde. Dependiendo de la configuración del usuario, se podrán establecer niveles altos (1 lógico) y bajos (0 lógico) de tensión. La señal captada por los sensores puede ser enviada directamente a un circuito de control, como un microcontrolador o una placa Arduino, para ser procesada según las necesidades específicas del usuario o la aplicación. [4]

El sensor requiere un voltaje de funcionamiento que oscila entre 3.3 V y 5 V, y presenta un ángulo de detección de 35°. Los terminales de conexión utilizadas incluyen D0, que corresponde a la señal recibida; GND, que es la tierra; y VCC, para la alimentación. [4]

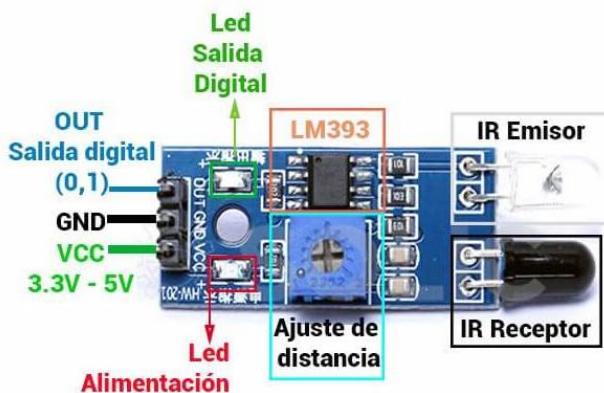


Figura 5: Diagrama del sensor de proximidad FC-51[6]

I2C 1602 LCD

El módulo I2C LCD1602 es un dispositivo que permite mostrar texto y caracteres en una pantalla LCD de 16x2 (16 columnas y 2 filas) utilizando el protocolo I2C [7]. Este módulo es útil para proyectos basados en Arduino, ya que facilita la visualización de información como lecturas

de sensores, mensajes o menús. Integra un chip PCF8574 que convierte los datos seriales del protocolo I2C en datos paralelos compatibles con la pantalla LCD.

El I2C LCD1602 combina un LCD convencional y un módulo I2C montado en la parte posterior de la pantalla. Este módulo I2C expande los puertos de entrada y salida del Arduino mediante el protocolo I2C, que utiliza únicamente dos cables: SDA (datos seriales) y SCL (reloj serial). Esto permite la comunicación simultánea entre varios dispositivos con direcciones únicas, optimizando el uso de recursos.

El módulo I2C convierte las señales del Arduino en comandos para controlar la pantalla LCD. La pantalla está compuesta por 16x2 celdas, cada una formada por una matriz de puntos 5x8 que se encienden o apagan según el voltaje aplicado, permitiendo mostrar diferentes caracteres o símbolos.

La dirección por defecto del módulo es 0x27, aunque en algunos casos puede ser 0x3F. Esta dirección puede modificarse alterando los pines A0, A1 y A2 mediante un puenteado. Por defecto, los pines A0, A1 y A2 están configurados como 1, pero al aplicar un puente, su valor cambia a 0.

El módulo también permite ajustar la retroiluminación y el contraste. La retroiluminación puede habilitarse o deshabilitarse mediante un puente. Además, un potenciómetro ubicado en la parte posterior permite ajustar el contraste, definiendo la claridad del texto mostrado. Girar el potenciómetro en sentido horario incrementa el contraste, mientras que girarlo en sentido antihorario lo reduce.

Librerías de Arduino IDE

- **BlinkSimpleEsp32.h:** Con esta librería permite conectar el modulo ESP32 a la plataforma Blynk. La cual proporciona funciones para conectar el ESP32 a la red, enviar y recibir datos entre el dispositivo y la aplicación Blynk que permite controlar diversos componentes del hardware a través de la interfaz.
- **Wire.h:** Esta librería se utiliza para la comunicación I2C, protcolo que permite comunicar serialmente dos cables que conectan múltiples dispositivos (sensores o pantallas) a un mismo bus. Permite enviar y recibir datos a través de I2C, facilitando la interacción con componentes externos que usan este protocolo.
- **LiquidCrystal_I2C.h:** Esta librería es específica para controlar pantallas LCD que utilizan el protocolo I2C. Es una forma simplificada de trabajar con pantallas LCD, ya que no necesita manejar múltiples pines para la comunicación, solo dos pines (SCL y SDA) para el I2C. A través de esta librería, se puede escribir texto en una pantalla LCD conectada al ESP3.
- **ESP32Servo.h:** Esta librería proporciona funciones para controlar la posición del servomotor a través de señales PWM (modulación de ancho de pulso), facilitando el control preciso de su movimiento.

Blynk

Blynk es una plataforma en la nube para IoT (Internet de las Cosas) diseñada para simplificar la creación y gestión de aplicaciones móviles y web personalizables para dispositivos IoT. Su objetivo es ofrecer una solución integral para gestionar todo el ciclo de vida de los dispositivos, desde el prototipo hasta la producción, garantizando seguridad en cada etapa. Blynk permite a los desarrolladores crear aplicaciones móviles de alta calidad, gestionar flotas de dispositivos, realizar pruebas de campo, y desplegar de manera eficiente, todo mientras se

facilita la administración de datos, configuraciones, actualizaciones y análisis de rendimiento [9].

Lista de componentes

Seguidamente, se presenta la lista de componentes necesarios para el desarrollo del parqueo junto con sus costos

Componente	Cantidad	Precio (₡)
ESP32	1	8800
Protoboard	1	7832
Kit de cables	1	2022
Servomotores SG90	2	4464
Sensor de proximidad FC-51	5	2284
Cable USB a micro USB	1	2300
I2C module	1	1615
Pantalla LCD	1	3900
Total		33217

Tablas 2: Componentes del proyecto con precio

Diagrama de flujo del proyecto

A continuación se muestra el diagrama de flujo del parqueo inteligente que indica el flujo del funcionamiento del mismo:



Figura 6: Diagrama de flujo del parqueo inteligente

Diagrama de bloques del proyecto

En la siguiente figura se muestra el diagrama de bloques del parqueo inteligente donde se especifican las señales de entrada y de salida:

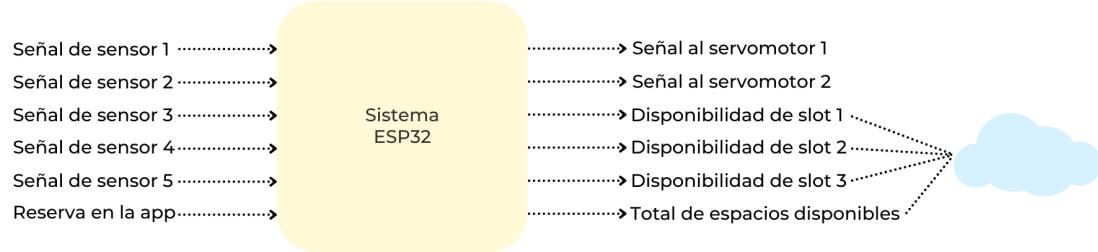


Figura 7: Diagrama de bloques del parqueo inteligente

Desarrollo

En este proyecto se implementó un sistema de parqueo inteligente utilizando un microcontrolador ESP32, sensores infrarrojos, servomotores, una pantalla LCD y la plataforma Blynk para la gestión remota y visualización de datos. Este sistema permite monitorear en tiempo real la disponibilidad de espacios, automatizar el control de las agujas en la entrada y salida, y reservar espacios de parqueo.

Componentes y Funcionalidad

■ Sensores infrarrojos:

- Tres sensores fueron utilizados para detectar la presencia de vehículos en cada uno de los espacios de parqueo.
- Dos sensores adicionales detectaron la entrada o salida de vehículos, activando el servomotor correspondiente para mover la aguja de entrada o salida.

■ Servomotores:

- Controlan las agujas de entrada y salida, subiendo o bajando en respuesta al estado detectado por los sensores.

■ Pantalla LCD:

- Muestra en tiempo real el estado de cada espacio de parqueo y el número total de espacios disponibles.

■ Blynk:

- Permite monitorear la disponibilidad de los espacios a través de una aplicación móvil.
- Incluye un switch para reservar espacios de forma remota.

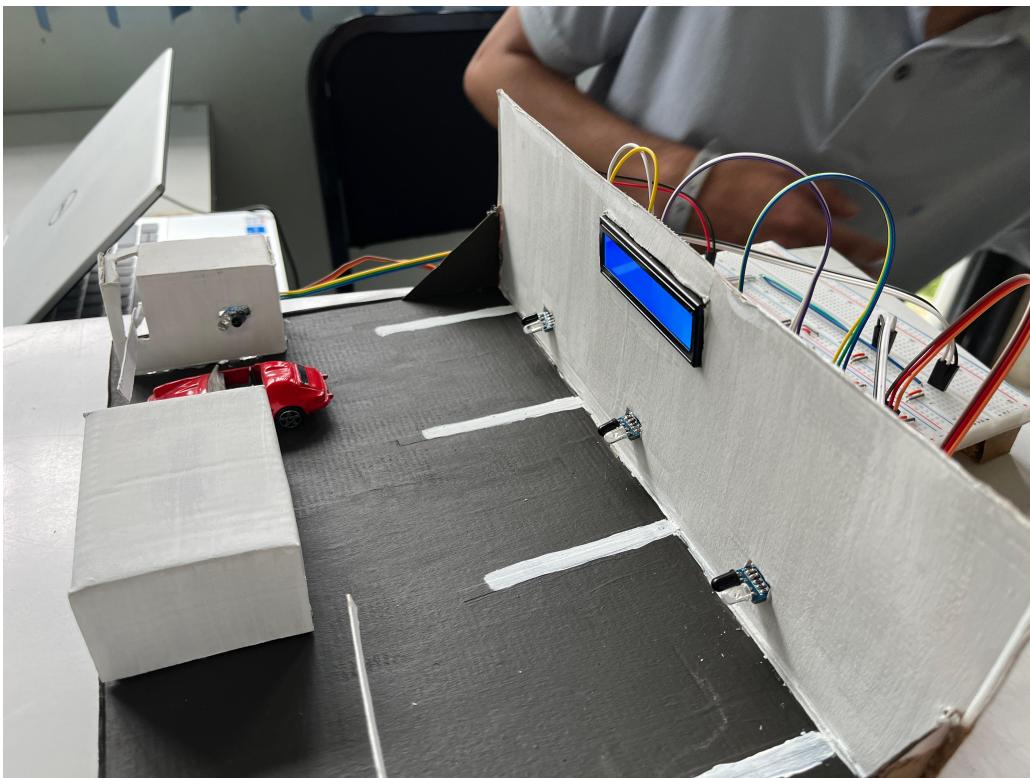


Figura 8: Maqueta del sistema de parqueo inteligente

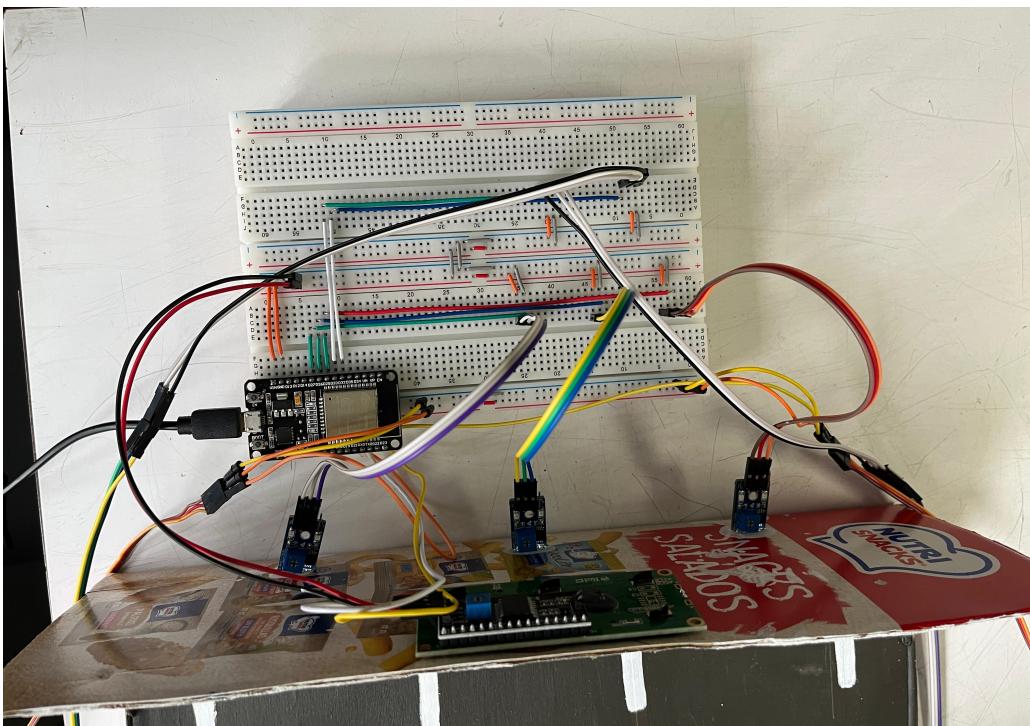


Figura 9: Circuito en protoboard del sistema

Funcionamiento del Sistema

El sistema opera de manera coordinada entre hardware y software para gestionar los espacios de parqueo de forma automatizada y remota. A continuación, se detalla su funcionamiento:

Inicialización del Sistema

El microcontrolador ESP32 se enciende y configura los periféricos:

- Los sensores infrarrojos se inicializan para detectar la presencia de vehículos y la entrada/salida.
- Los servomotores se posicionan en “cerrado” para las agujas.
- La pantalla LCD muestra el estado inicial de los espacios.
- Se establece la conexión Wi-Fi para comunicarse con la plataforma Blynk.

Gestión de Entrada de Vehículos

1. **Detección de entrada:** El sensor infrarrojo en la entrada detecta la aproximación de un vehículo. El ESP32 envía una señal al servomotor de la entrada para que suba la aguja.
2. **Cierre de la aguja:** Una vez que el vehículo pasa completamente, el sensor de entrada indica al servomotor que baje la aguja.
3. **Actualización de disponibilidad:** Los sensores en los espacios detectan la ocupación de un slot. El microcontrolador actualiza el estado a “no disponible” (NA) y reduce el contador total de espacios disponibles. Esta información se refleja en la pantalla LCD y en la aplicación móvil.

Gestión de Salida de Vehículos

1. **Liberación de espacio:** Cuando el vehículo abandona un espacio, el sensor correspondiente detecta que está vacío. El sistema actualiza el estado del espacio a “disponible” (A) y aumenta el contador total de espacios disponibles.
2. **Detección de salida:** El sensor infrarrojo en la salida detecta el movimiento de un vehículo. El ESP32 envía una señal al servomotor de la salida para que suba la aguja.
3. **Cierre de la aguja:** El servomotor baja la aguja una vez que el vehículo ha salido por completo.

Visualización en la Pantalla LCD

La pantalla LCD muestra en tiempo real el estado de los espacios de parqueo en el siguiente formato:

- Slot1:A, Slot2:A, Slot3:A, Total:3 (cuando todos los espacios están disponibles).
- Slot1:NA, Slot2:A, Slot3:NA, Total:1 (si solo el segundo espacio está disponible, por ejemplo).

La información se actualiza dinámicamente cada vez que un sensor detecta un cambio en el estado.

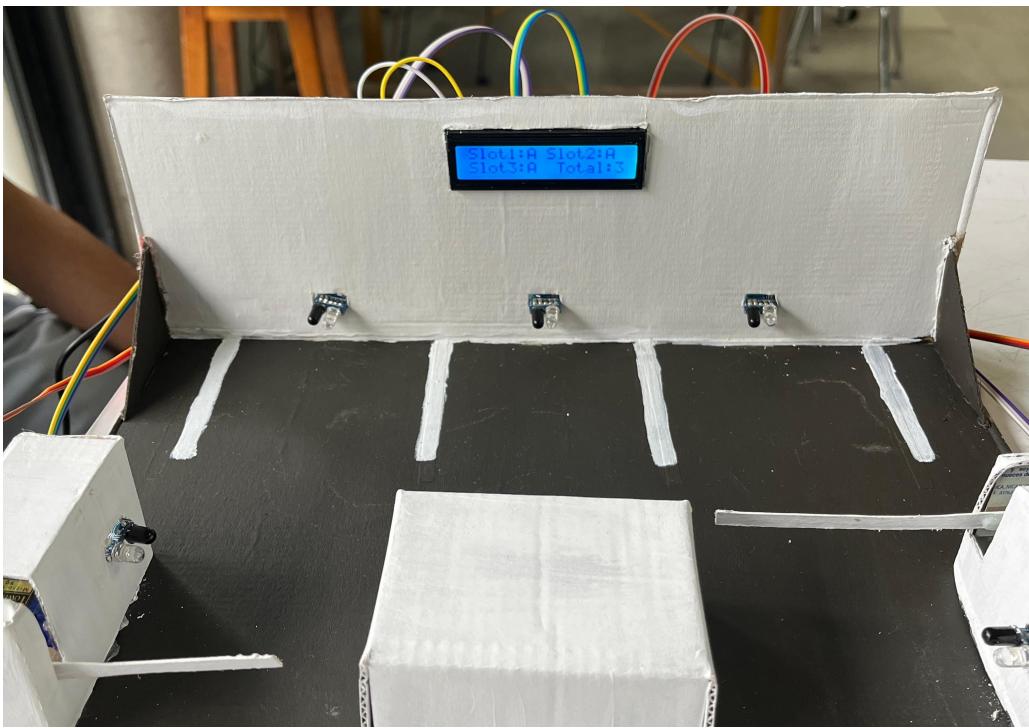


Figura 10: Visualización del estado de disponibilidad de los espacios en la pantalla LCD

Control y Monitoreo Remoto con Blynk

- **Monitoreo:** La aplicación móvil Blynk refleja el estado actual del parqueo, mostrando la disponibilidad de cada espacio (0 para disponible, 1 para ocupado) y el total de espacios libres.
- **Reserva de espacios:** La aplicación incluye un switch que permite a los usuarios reservar un espacio. Cuando se activa, el sistema marca el espacio como “reservado” incluso si está vacío.

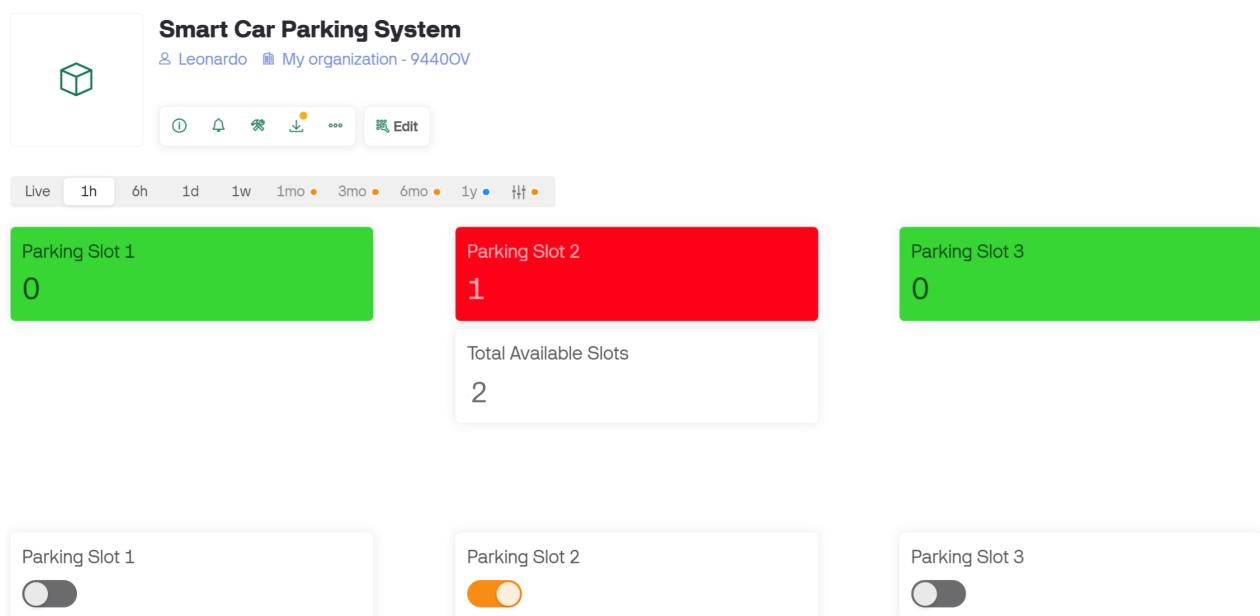


Figura 11: Visualización de espacios disponibles en la plataforma Blynk por conexión WiFi cuando el slot 2 está ocupado

Análisis de Resultados

Los resultados obtenidos durante la implementación y pruebas del sistema de parqueo inteligente fueron satisfactorios, demostrando la efectividad de la integración entre los componentes de hardware y software. Los sensores infrarrojos presentaron un alto grado de precisión en la detección de vehículos en los espacios de parqueo y en las entradas/salidas, logrando una tasa de acierto sumamente alta. Sin embargo, se observó que en condiciones de iluminación ambiental intensa, la sensibilidad de los sensores disminuyó significativamente, lo que en algunos casos llevó a lecturas incorrectas. Este aspecto resalta la necesidad de considerar tecnologías complementarias para garantizar un desempeño consistente en diversas condiciones.

La sincronización entre el microcontrolador ESP32, la pantalla LCD y la plataforma Blynk se llevó a cabo de manera eficiente, permitiendo la actualización en tiempo real del estado de los espacios de parqueo tanto en la pantalla local como en la aplicación móvil. Los tiempos de respuesta en la aplicación fueron de aproximadamente 2 segundos, evidenciándose un retraso significativo en la actualización de datos en la plataforma Blynk. Por otro lado, la funcionalidad de reserva remota demostró ser útil y práctica.

En términos de automatización, el sistema logró operar sin intervención manual, controlando de manera precisa las agujas en las entradas y salidas mediante los servomotores, que respondieron rápidamente a las señales del ESP32. Esto significa una gran mejora en la experiencia del usuario al ingresar o salir del parqueo.

Conclusiones y recomendaciones

El sistema desarrollado cumplió con el objetivo de automatizar la detección y gestión de espacios de parqueo. La integración de sensores infrarrojos, servomotores y la pantalla LCD permitió un monitoreo local preciso, mientras que la conectividad con la plataforma Blynk ofreció una experiencia de usuario moderna y accesible. Esto demuestra que la implementación de tecnologías IoT puede optimizar significativamente la operación de estacionamientos, eliminando la necesidad de supervisión o accionamiento manual constante.

Se logró una integración con la aplicación Blynk que permite a los usuarios consultar la disponibilidad de espacios y realizar reservas en tiempo real desde cualquier lugar con conexión a Internet. Este aspecto no solo mejora la comodidad de los usuarios, sino que también evidencia la robustez del sistema en términos de comunicación y sincronización entre hardware y software.

Aunque el sistema funcionó correctamente, se identificaron algunas limitaciones, como la sensibilidad de los sensores infrarrojos a condiciones de luz ambiental extrema y los retrasos en la actualización de datos en la plataforma Blynk debido a las condiciones de la conexión Wi-Fi. Estas áreas representan oportunidades para mejorar la robustez y el desempeño general del sistema.

Para superar las limitaciones de los sensores infrarrojos, se recomienda explorar el uso de tecnologías complementarias como sensores ultrasónicos o cámaras con algoritmos de visión artificial. Estas herramientas pueden aumentar la precisión en la detección de vehículos, especialmente en condiciones de iluminación desfavorables.

Por otro lado, se recomienda optar por una pantalla LCD que ya cuente con el módulo I2C integrado, ya que adquirir ambos componentes por separado y soldar sus pines puede resultar más complejo, requiriendo de mayor tiempo y las herramientas adecuadas, además de aumentar el riesgo de errores durante el montaje, como conexiones inadecuadas o daños en los componentes.

Referencias

- [1] AWS. (s.f.). ¿Qué es IoT (Internet de las cosas)? . Recuperado de <https://aws.amazon.com/es/what-is/iot/>
- [2] Espressif Systems. (2019). ESP32 Series Datasheet (Version 3.0). Recuperado de <https://www.alldatasheet.com/datasheet-pdf/view/1148023/ESPRESSIF/ESP32.html>
- [3] ETC. (s.f.). SG90 9g Micro Servo. Recuperado de <https://www.alldatasheet.es/datasheet-pdf/view/1572383/ETC/SG90.html>
- [4] Mactronica. (2024). Sensor de obstáculos infrarrojo. Recuperado de <https://www.mactronica.com.co/sensor-de-obstaculos-infrarrojo-fc-51>
- [5] AG Electrónica SAPI de CV. (2024). Sensor de proximidad infrarrojo FC-51. Recuperado de <https://agelectronica.lat/pdfs/textos/0/OKY3127.PDF>
- [6] UElectronics. (s.f.). FC-51 Sensor de Obstáculos Reflectivo Infrarrojo. Recuperado de from <https://uelectronics.com/producto/fc-51-sensor-de-obstaculos-reflectivo-infrarrojo/>
- [7] SunFounder. (s.f.). I2C LCD 1602. Recuperado de https://docs.sunfounder.com/projects/ultimate-sensor-kit/en/latest/components_basic/21-component_i2c_lcd1602.html
- [8] Handson Technology. (s.f.). I2C Serial Interface 1602 LCD Module. Recuperado de https://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf
- [9] Blynk. (s.f.). Blynk: The platform for the Internet of Things. Recuperado de <https://blynk.io/>

Anexos

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-D0WDQ6, ESP32-D0WD, ESP32-D2WD, and ESP32-S0WD. For details on part numbers and ordering information, please refer to [Part Number and Ordering Information](#).

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 and 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

1.6 Block Diagram

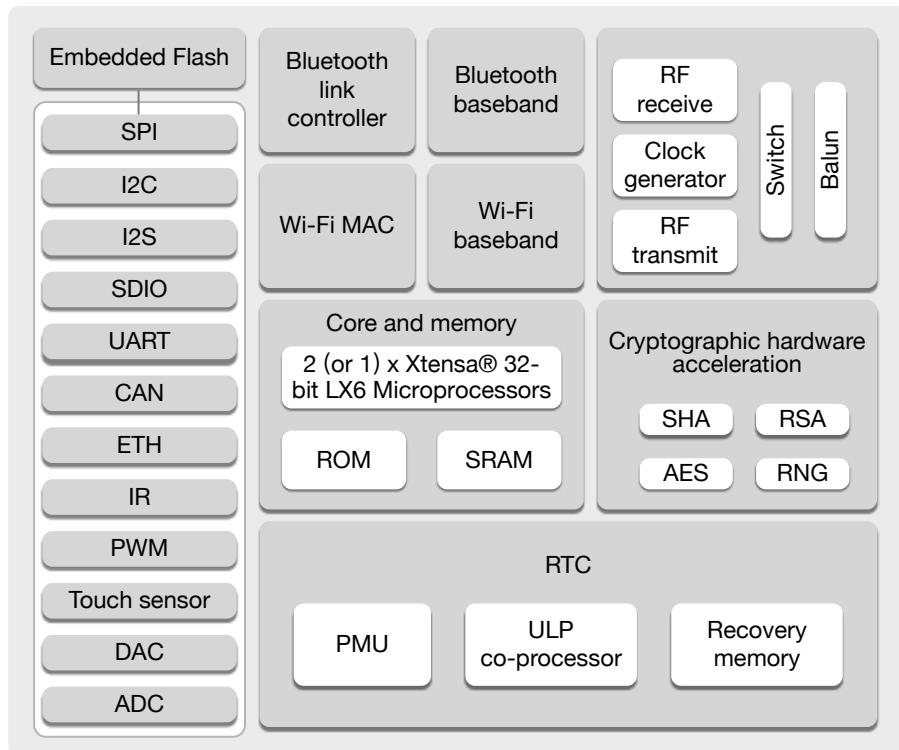


Figure 1: Functional Block Diagram

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 [Part Number and Ordering Information](#).

2. Pin Definitions

2.1 Pin Layout

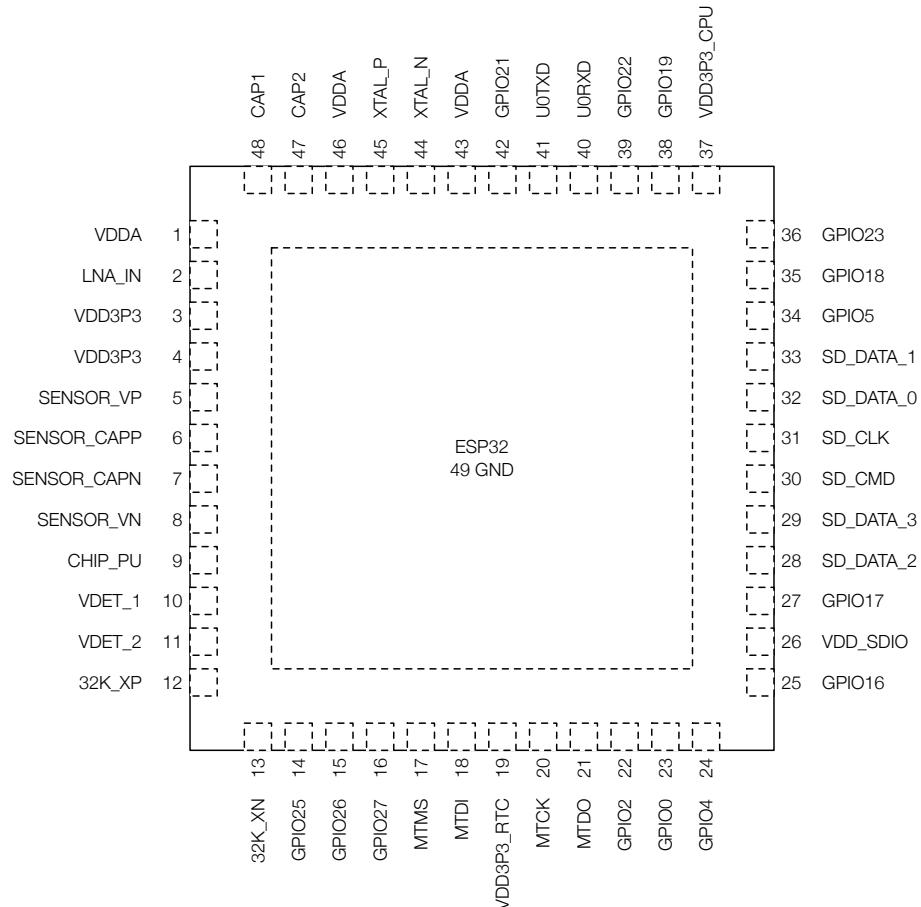


Figure 2: ESP32 Pin Layout (QFN 6*6, Top View)

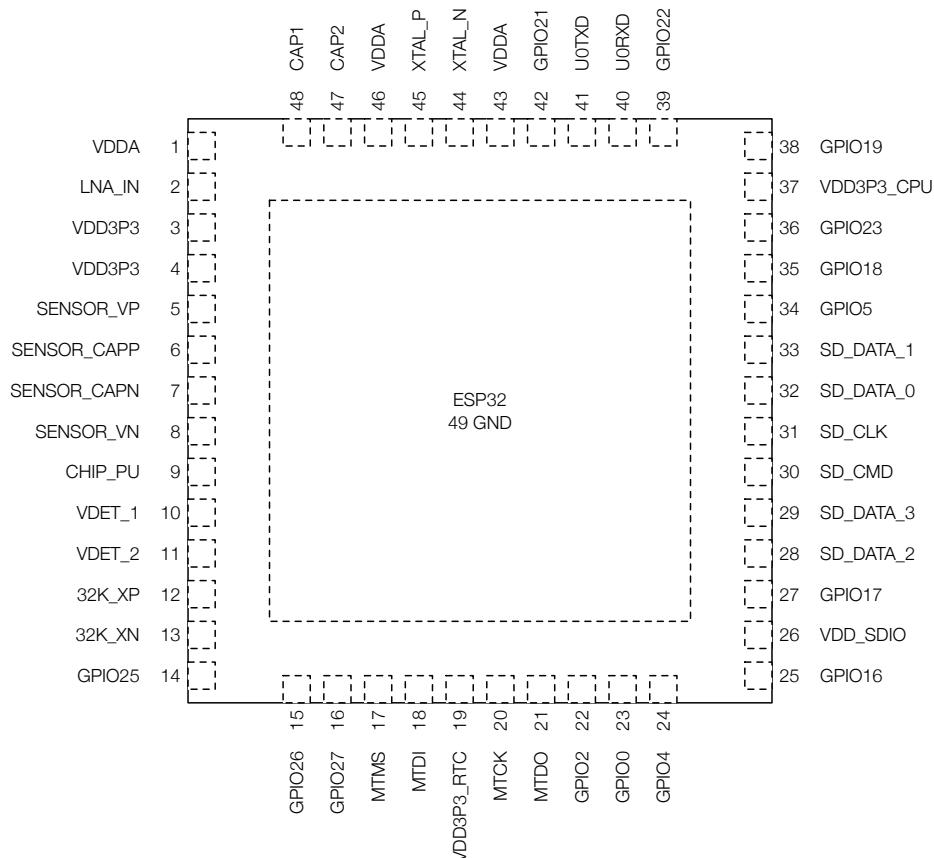


Figure 3: ESP32 Pin Layout (QFN 5*5, Top View)

Note:

For details on ESP32's part numbers and the corresponding packaging, please refer to [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.

Name	No.	Type	Function			
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		
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, SPIID		
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 nF capacitor and 20 kΩ resistor in parallel to CAP1			
CAP1	48	I	Connects to a 10 nF series capacitor to ground			
GND	49	P	Ground			

Note:

- ESP32-D2WD's pins GPIO16, GPIO17, SD_CMD, SD_CLK, SD_DATA_0 and SD_DATA_1 are used for connecting the embedded flash, and are not recommended for other uses.
- 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](#).
- In most cases, the data port connection between the ESP32 and external flash is as follows: SD_DATA0/SPIQ = IO1/DO, SD_DATA1/SPIID = IO0/DI, SD_DATA2/SPIHD = IO3/HOLD#, SD_DATA3/SPIWP = IO2/WP#.

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:

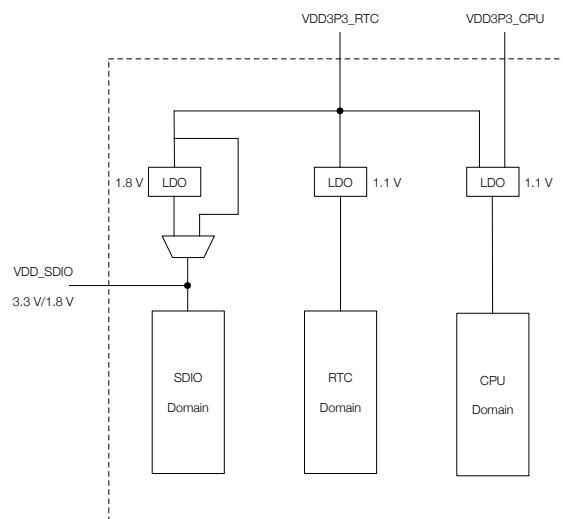


Figure 4: ESP32 Power Scheme

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.

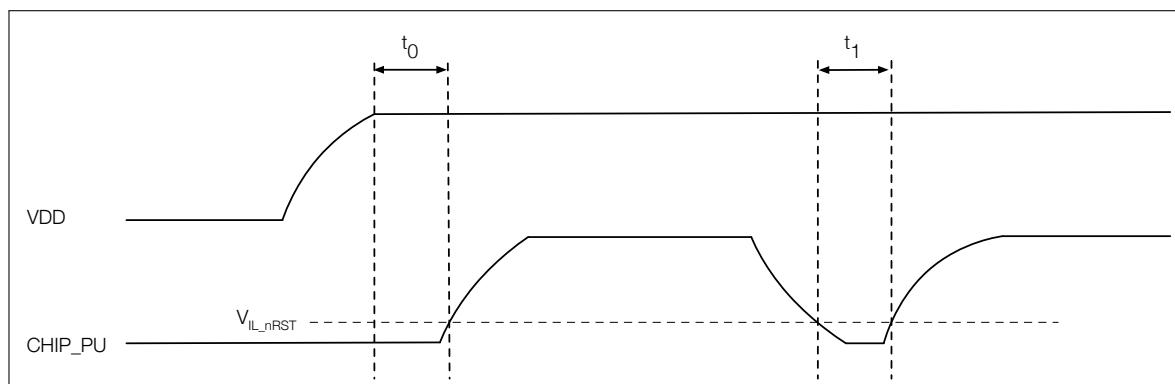


Figure 5: ESP32 Power-up and Reset Timing

SENSOR DE PROXIMIDAD INFRARROJO FC-51

OKY3127



Productos evaluados por ingenieros calificados



Garantía y seguridad en cada producto



Experiencia de compra en la calidad como sello distintivo

Descripción

El OKY3127 es un sensor de proximidad infrarrojo que basa su principio de funcionamiento en un transmisor y receptor IR para identificar obstáculos delante del sensor en un rango de 2 a 30cm de distancia. El módulo cuenta con un potenciómetro que permite al usuario ajustar el rango de detección.

El sensor cuenta con una respuesta estable con luz ambiente o en completa oscuridad, se puede utilizar con Arduino, Raspberry Pi entre otras tarjetas y microcontroladores que se alimenten en un rango de 3.3VDC a 5VDC.

Modo de funcionamiento

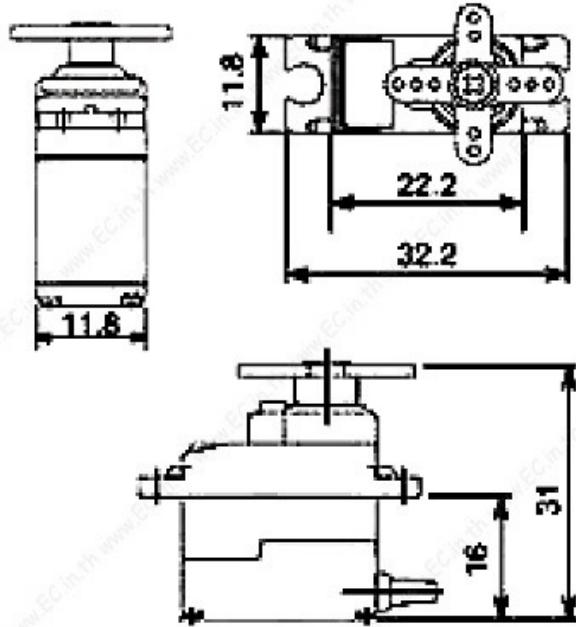
- Cuando el módulo detecta un obstáculo frente a la señal, el led verde enciende a nivel de la placa, mientras que el puerto de salida OUT emite una señal continua de bajo nivel. El módulo puede detectar distancias de 2 a 30 cm, tiene un ángulo de detección de 35°. La distancia de detección se puede ajustar girando el potenciómetro: en el sentido de las agujas del reloj para aumentar la distancia de detección y en sentido contrario para reducirla.

- Los sensores infrarrojos activos detectan la luz reflejada, por lo que la reflectividad y la forma del objetivo son críticas para el rango de detección. La distancia mínima de detección se da con objetos negros y la máxima con objetos blancos. Objetos pequeños con áreas pequeñas se detectan a distancias más cortas, mientras que objetos grandes con áreas grandes se detectan a mayores distancias.
- El puerto de salida OUT se puede conectar directamente al puerto IO de un microcontrolador y puede conducir directamente un relevador de 5V. Conexiones: VCC a VCC, GND a GND, OUT a IO.
- Puede ser alimentado con un voltaje de 3-5V DC. Cuando se enciende, el indicador de encendido rojo se ilumina.

Especificaciones

Parámetro	Descripción
Voltaje de trabajo	3 V~5V
Angulo de cobertura	35°
Rango de detección	2cm~30cm
Consumo de corriente	23mA (3.3V) 43mA (5V)
Circuito de detección	basado en el LM393 y tecnología IR
Led indicador	Alimentación y obstáculos
Sensibilidad	Ajustable
Orificio de montaje	Diámetro 3mm
Dimensiones	31mmx14mm

SG90 9 g Micro Servo



Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but *smaller*. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

Specifications

- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10 μ s
- Temperature range: 0 °C – 55 °C



I2C Serial Interface 1602 LCD Module

This is I2C interface 16x2 LCD display module, a high-quality 2 line 16 character LCD module with on-board contrast control adjustment, backlight and I2C communication interface. For Arduino beginners, no more cumbersome and complex LCD driver circuit connection. The real significance advantages of this I2C Serial LCD module will simplify the circuit connection, save some I/O pins on Arduino board, simplified firmware development with widely available Arduino library.



SKU: [DSP-1182](#)

Brief Data:

- Compatible with Arduino Board or other controller board with I2C bus.
- Display Type: Negative white on Blue backlight.
- I2C Address: 0x38-0x3F (0x3F default)
- Supply voltage: 5V
- Interface: I2C to 4bits LCD data and control lines.
- Contrast Adjustment: built-in Potentiometer.
- Backlight Control: Firmware or jumper wire.
- Board Size: 80x36 mm.

Setting Up:

Hitachi's HD44780 based character LCD are very cheap and widely available, and is an essential part for any project that displays information. Using the LCD piggy-back board, desired data can be displayed on the LCD through the I2C bus. In principle, such backpacks are built around PCF8574 (from NXP) which is a general purpose bidirectional 8 bit I/O port expander that uses the I2C protocol. The PCF8574 is a silicon CMOS circuit provides general purpose remote I/O expansion (an 8-bit quasi-bidirectional) for most microcontroller families via the two-line bidirectional bus (I2C-bus). Note that most piggy-back modules are centered around PCF8574T (SO16 package of PCF8574 in DIP16 package) with a default slave address of 0x27. If your piggy-back board holds a PCF8574AT chip, then the default slave address will change to 0x3F. In short, if the piggy-back board is based on PCF8574T and the address connections (A0-A1-A2) are not bridged with solder it will have the slave address 0x27.



Address selection pads in the I2C-to-LCD piggy-back board.

Table 5. PCF8574A address map

Pin connectivity			Address of PCF8574A								Address byte value		7-bit hexadecimal address without R/W
A2	A1	A0	A6	A5	A4	A3	A2	A1	A0	R/W	Write	Read	
V _{SS}	V _{SS}	V _{SS}	0	1	1	1	0	0	0	-	70h	71h	38h
V _{SS}	V _{SS}	V _{DD}	0	1	1	1	0	0	1	-	72h	73h	39h
V _{SS}	V _{DD}	V _{SS}	0	1	1	1	0	1	0	-	74h	75h	3Ah
V _{SS}	V _{DD}	V _{DD}	0	1	1	1	0	1	1	-	76h	77h	3Bh
V _{DD}	V _{SS}	V _{SS}	0	1	1	1	1	0	0	-	78h	79h	3Ch
V _{DD}	V _{SS}	V _{DD}	0	1	1	1	1	0	1	-	7Ah	7Bh	3Dh
V _{DD}	V _{DD}	V _{SS}	0	1	1	1	1	1	0	-	7Ch	7Dh	3Eh
V _{DD}	V _{DD}	V _{DD}	0	1	1	1	1	1	1	-	7Eh	7Fh	3Fh

Address Setting of PCD8574A (extract from PCF8574A data specs).

Note: When the pad A0~A2 is open, the pin is pull up to VDD. When the pin is solder shorted, it is pull down to VSS.

The default setting of this module is A0~A2 all open, so is pull up to VDD. The address is 3Fh in this case.

Reference circuit diagram of an Arduino-compatible LCD backpack is shown below. What follows next is information on how to use one of these inexpensive backpacks to interface with a microcontroller in ways it was exactly intended.