

Informe Integrador:

Proyecto Juego

Hecho por: Mariano López Fara

GitHub: <https://github.com/MarianoLF410/Integrador-Juego.git>

Mi proyecto consiste en un juego para desafiarte a ver si sabés qué son las palabras graves, agudas y esdrújulas.

El proyecto está planteado de una forma que sea:

- Fácil de usar
 - Portatil
 - Rejugable
-

Sensores utilizados:

El sensor que utilizo es un encoder rotatorio EC11 de 20 pulsos por rotación con pulsador. El encoder genera dos señales cuadradas (en los canales A y B) que determinan el sentido de giro y el número de pasos.

Características:

- Tipo: Encoder incremental y absoluto.
- Número de Pulsos: 20 pulsos
- Vida Útil: 15,000 ciclos para la mayoría de los modelos.
- Torque: 6 mN·m.
- Switch de Presión: Con función de switch de presión (viaje de 1.5 mm).
- Voltaje de Operación: 5V DC.
- Corriente de Operación: 10 mA.
- Conectividad: Terminales de salida A y B, con señal proporcional a la diferencia de fase.
- Especificaciones Adicionales
 - Temperatura de Operación: -40 °C a +85 °C.
 - Rango de Humedad: 90-95% RH a 60 °C.
 - Resistencia a la Insulación: Mínimo 100 MΩ a 250V DC.
 - Protección: A prueba de voltaje de 300V AC durante 1 minuto.

Principio de funcionamiento:

El encoder produce dos señales cuadradas (canales A y B) desfasadas 90 grados entre sí ([Código Gray de 2 bits](#)). Este desfase permite identificar la dirección de rotación: Sentido horario (CW): A lidera a B. Sentido antihorario (CCW): B lidera a A. La señal se genera al girar el eje, y cada ciclo completo (un paso A y B) representa un pulso.

Actuadores utilizados:

Buzzer:

Es un componente electroacústico diseñado para emitir sonidos o tonos continuos. Este buzzer específico es del tipo electromecánico con un circuito de manejo interno.

Existen dos tipos de buzzers:

Buzzer Activo:

- Este tipo de buzzer incluye un circuito interno que genera el sonido automáticamente cuando recibe corriente.

Buzzer Pasivo:

- Este tipo de buzzer requiere una señal PWM para generar tonos o sonidos, ya que no tiene circuito interno.

En mi caso mi buzzer es del tipo activo.

Características:

- Voltaje Nominal: 6V DC
- Voltaje de Operación: 4 a 8 V DC
- Corriente Nominal: $\leq 30\text{mA}$
- Salida de Sonido a 10cm: $\geq 85\text{dB}$
- Frecuencia Resonante: $2300 \pm 300\text{Hz}$
- Tono: Continuo
- Temperatura de Operación: -25°C a $+80^{\circ}\text{C}$
- Temperatura de Almacenamiento: -30°C a $+85^{\circ}\text{C}$
- Peso: 2g

LCD:

Este es un tipo de pantalla de cristal líquido que puede mostrar hasta 20 caracteres por fila y 4 filas de texto. Como utiliza comunicación I2C, es capaz de conectar la pantalla a un microcontrolador utilizando solo dos cables para la transmisión y recepción de datos.

Características:

- Tipo de Display: LCD de 20 caracteres por 4 líneas.
- Compatibilidad: Compatible con placas Arduino y otros controladores con bus I2C.
- Tipo de Visualización: Texto blanco sobre fondo azul.
- Dirección I2C: Rango de 0x38 a 0x3F (0x3F por defecto).
- Voltaje de Suministro: 5V.
- Interfaz: Conversión de datos y líneas de control de 4 bits a través de I2C.

- Ajuste de Contraste: Potenciómetro integrado.
 - Control de Retroiluminación: Controlado por firmware o mediante cables de puente.
 - Dimensiones de la Placa: 98 mm x 60 mm.
-

Otros componentes:

LM7805

Es un regulador de voltaje lineal de tres terminales que proporciona una salida constante de 5V.

Características:

- Voltaje de Entrada (VI)
 - Para VO = 5V a 18V: 35V
 - Para VO = 24V: 40V
- Corriente de Salida (IO)
 - Hasta 1A
- Voltaje de Salida (VO)
 - 4.75V a 5.25V (con $5.0\text{mA} \leq IO \leq 1.0\text{A}$, $PO \leq 15\text{W}$)
 - Típicamente 5.0V a 25°C
- Regulación de Línea (Regline)
 - VI = 8V a 12V: -1.6 mV a 50 mV
- Regulación de Carga (Regload)
 - IO = 5mA a 1.5A: -9 mV a 100 mV
 - IO = 250mA a 750mA: -4 mV a 50 mV
- Caída de Voltaje (VDrop)
 - IO = 1A, TJ = +25°C: -2V
- Rango de Temperatura
 - Temperatura de Operación (TOPR): 0°C a +125°C

Principio de funcionamiento:

El LM7805 regula el voltaje de entrada utilizando un transistor interno y un sistema de control de retroalimentación. El dispositivo mantiene la salida en 5V, ajustando automáticamente la corriente que pasa a través del transistor, de modo que la diferencia de voltaje entre la entrada y la salida se "descarta" como calor.

Level Shifter:

Es un dispositivo electrónico utilizado para adaptar los niveles de voltaje entre dos sistemas que operan a diferentes voltajes. Su función principal es permitir la comunicación entre dispositivos que trabajan con distintas tensiones lógicas, como, en mi caso, un microcontrolador de 3.3V y una pantalla LCD de 5v.

Características:

- Voltajes de Suministro:
 - Bajo Voltaje (LV): 1.5 V a 7 V
 - Alto Voltaje (HV): LV a 18 V
- Canales: 4 canales bidireccionales
- Dimensiones: Tamaño: 0.4" × 0.5" × 0.08" (13 mm × 10 mm × 2 mm)
- Conexiones:
 - Compatible con breadboards
- Resistencias Pull-Up:
 - Se utilizan resistencias de 10 kΩ para cada canal, que ayudan a lograr tiempos de subida adecuados para señales rápidas (hasta 400 kHz para I²C). En mi caso utilizo la resistencia PULL-UP del ESP32.
- Interfaz:
 - Compatible con interfaces digitales rápidas (I²C, SPI, TTL asíncrono).

Funcionamiento:

El principio básico de operación de un Level Shifter se basa en el uso de transistores y resistencias para adaptar los niveles de voltaje de las señales digitales.

ESP32:

El ESP32 es un microcontrolador de 32 bits, desarrollado por Expressif Systems, que incluye capacidades de conectividad como Wi-Fi y Bluetooth (tanto Classic como Low Energy). Es ampliamente utilizado en proyectos de Internet de las Cosas (IoT) debido a su bajo costo, bajo consumo de energía y alta capacidad de procesamiento.

Características:

- Módulo: ESP-32S
- Tipo: MCU Wi-Fi-BT-BLE
- CPU y Memoria Interna Microprocesadores: 2 núcleos Xtensa® 32-bit LX6
- Memoria: 448 KBytes ROM - 520 KBytes SRAM - 8 KBytes SRAM en RTC (accesible durante el modo de sueño profundo) - 1 Kbit de EFUSE (configuración del sistema y aplicaciones del cliente) -
- Conectividad
 - Wi-Fi: 802.11 b/g/n/d/e/i/k/r (hasta 150 Mbps)
 - Bluetooth: v4.2 BR/EDR y BLE
 - Rango de frecuencia: 2.4GHz - 2.5GHz
- Consumo de Energía
 - Modos de potencia:
 - Activo: 160 - 260 mA
 - Modem-sleep: 5 - 20 mA
 - Light-sleep: 0.8 mA
 - Deep-sleep: 0.15 mA
- Hiberna
- Interfaz de Periféricos

- GPIO: 38 pines, incluyendo ADC, DAC, PWM, I2C, UART, I2S, y más.
 - Sensores: Sensores de toque capacitivos y amplificadores de bajo ruido.
 - Características del Radio
 - Wi-Fi:
 - Potencia de salida: 15.5 - 21.5 dBm
 - Sensibilidad: hasta -98 dBm
 - Bluetooth LE:
 - Sensibilidad: -98 dBm
 - Potencia de transmisión: hasta +10 dBm
-

SoftWare

Visual Studio Code:

Es un editor de código fuente ligero, rápido y de código abierto, desarrollado por Microsoft. Aunque no es un IDE completo por sí mismo, se puede extender fácilmente mediante complementos y configuraciones para convertirse en una herramienta potente para la programación. Está diseñado para ser altamente configurable y soporta una amplia variedad de lenguajes de programación, lo que lo convierte en una de las opciones más populares entre desarrolladores de software, tanto para proyectos pequeños como para proyectos más grandes y complejos.

Características:

- Ligero y Rápido: A diferencia de otros entornos de desarrollo integrados más pesados, VS Code es muy rápido y ligero, lo que lo hace adecuado incluso para equipos con recursos limitados.
- Compatibilidad Multilenguaje: Soporta una amplia variedad de lenguajes de programación como JavaScript, Python, C/C++, Java, HTML/CSS, Ruby, PHP, Go, Rust, entre muchos otros, gracias a su sistema de extensiones.
- IntelliSense: Ofrece autocompletado de código y sugerencias inteligentes mientras se escribe, lo que acelera el proceso de desarrollo y reduce los errores de sintaxis.
- Depuración: Tiene una herramienta de depuración integrada que permite establecer puntos de interrupción, inspeccionar variables y recorrer el código paso a paso.
- Control de Versiones: Integra herramientas de control de versiones como Git, permitiendo a los desarrolladores gestionar repositorios directamente desde el editor, realizar
- Extensiones y Personalización: VS Code tiene un vasto mercado de extensiones que permite añadir soporte para diferentes lenguajes, temas, depuradores, y herramientas de integración. Las extensiones permiten agregar características como linters, formateadores de código, integración con bases de datos, entre otras.
- Terminal Integrado: Incluye una terminal que permite ejecutar comandos del sistema, scripts o programas directamente desde el entorno de desarrollo, sin necesidad de salir del editor.

- Refactorización y Linting: Proporciona herramientas para mejorar y refactorizar el código, además de integrar linters para garantizar que el código sigue las buenas prácticas y estándares de calidad.

Framework:

ESP-IDF

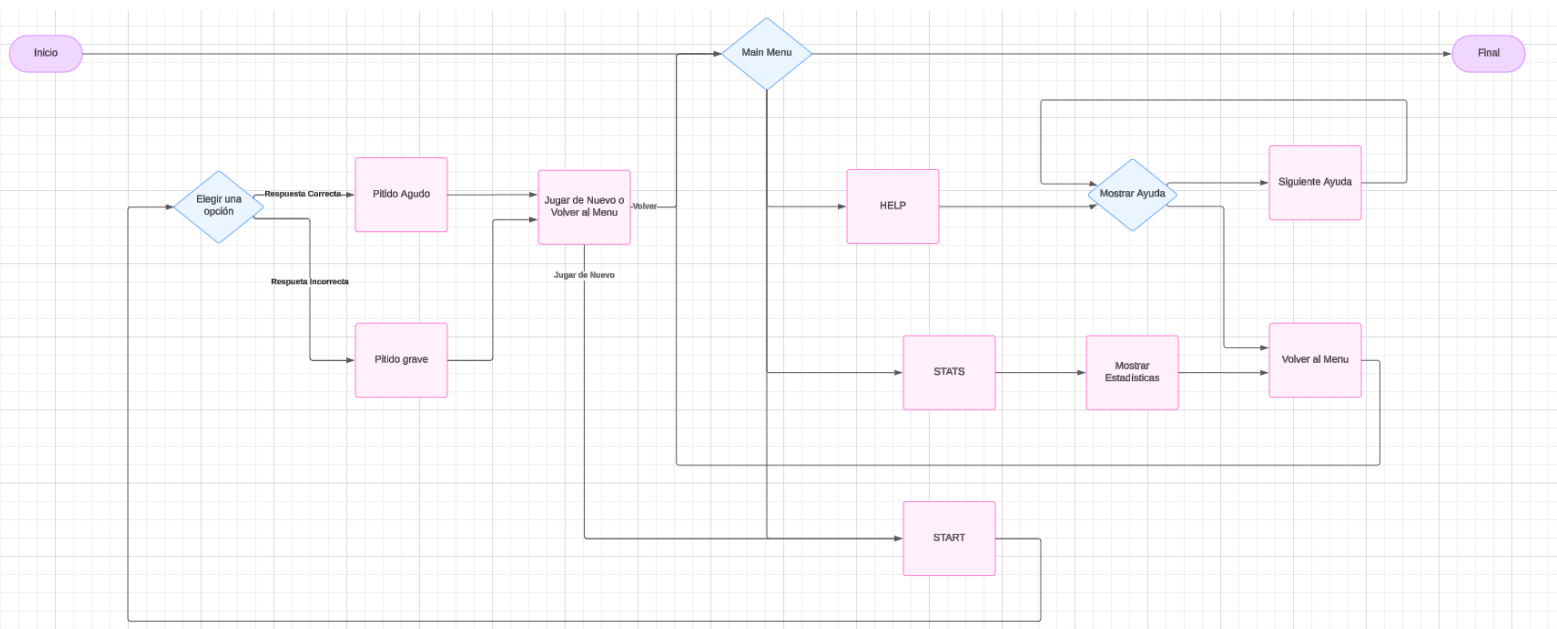
El ESP-IDF es el marco de desarrollo oficial proporcionado por Expressif Systems para programar los microcontroladores ESP32 y ESP8266. Este framework es de código abierto y está diseñado específicamente para el desarrollo de aplicaciones de Internet de las Cosas, utilizando las capacidades de estos microcontroladores.

El framework de ESPIDF se implementó a Visual Studio Code mediante PlatformIO.

Platformio es un entorno de desarrollo multiplataforma basado en una poderosa herramienta de construcción y un gestor de bibliotecas, que simplifica el proceso de desarrollo, especialmente en proyectos embebidos.

FirmWare:

Para la explicación del código, hice un diagrama de flujo para un entendimiento fácil:



Este diagrama de flujo explica el funcionamiento del proyecto de una forma mucho más sencilla comparado con el programa dentro de VSC.

PCB

Para este proyecto era necesario fabricar una PCB desde 0, por lo que para crearla hice lo siguiente:

- Diseñar la placa:
 - Para diseñar la placa utilice [KiCAD 8.0](#) utilizando algunas huellas de componentes externos que se encuentran para descargar en el GitHub.
- Imprimir la placa:
 - Utilice una impresora de tóner junto a una hoja de una revista para imprimir mi diseño.
- Planchar:
 - Luego de imprimir, planché el diseño a la placa de cobre para que se traspasara.
- Ácido:
 - Utilice ácido para comer las pistas de cobre innecesarias.
- Agujerear:
 - Utilice el taladro de la escuela para hacer los agujeros a la placa.
- Soldar:
 - Por último soldé los componentes a la placa.