

Requerimiento del Proyecto: Micro-CPU 2Bits

I. Objetivo del Proyecto

El objetivo de este proyecto es construir un modelo funcional y simplificado de una Unidad Central de Procesamiento (CPU) y un sistema de entrada/salida utilizando la placa RP2040 Zero.

El proyecto "**Micro-CPU 2Bits**" simulará:

1. El **Ciclo de Instrucción** (Fetch, Decode, Execute, Write-Back) mediante una secuencia de LEDs.
2. Las **Banderas de Estado** (Zero, Negative, Overflow) al realizar operaciones aritméticas.
3. Una calculadora básica de **2 bits** (operaciones con números del 0 al 3).

II. Hardware y Software Requerido

A. Listado de Partes (Materiales)

El proyecto se realizará en parejas o individualmente, y se deben utilizar los siguientes componentes:

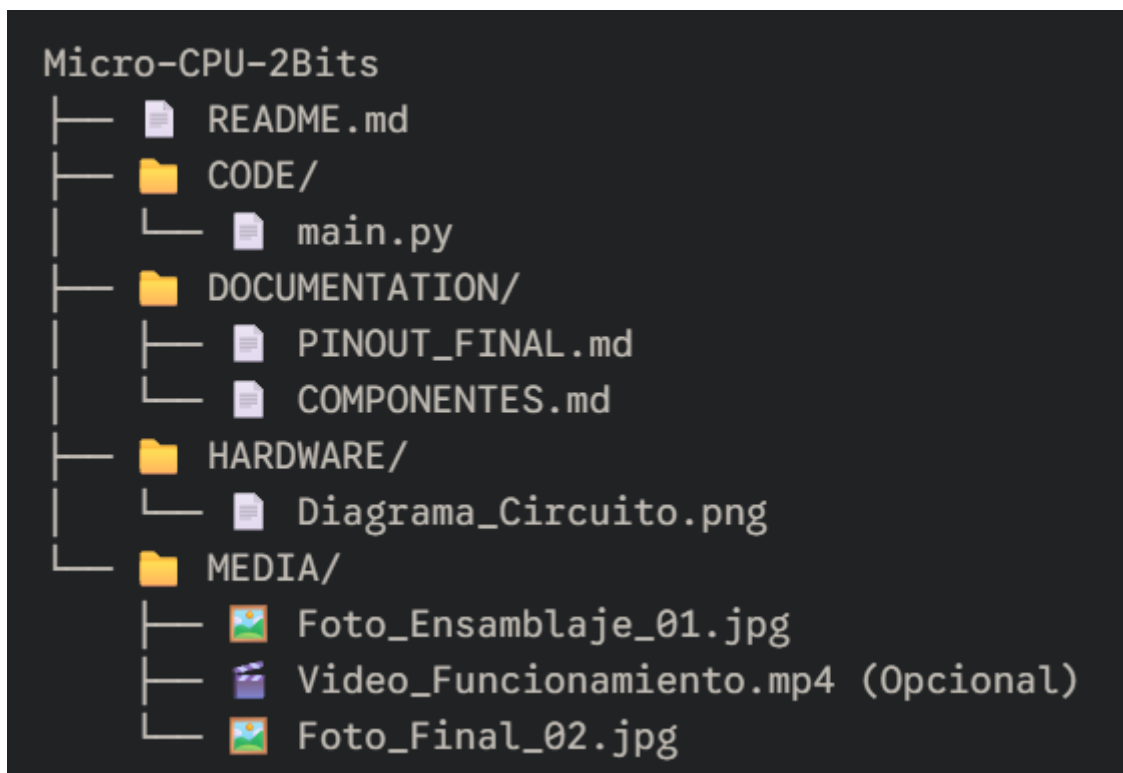
- 1x Placa de Desarrollo RP2040 Zero (u otra placa basada en RP2040 con soporte para MicroPython).
- 1x Protoboard.
- 7x Botones Pulsadores (4 para los dígitos 0-3 y 3 para las operaciones +, -, =).
- 6x LEDs Monocromáticos (para funciones Fetch, Decode, Execute, Write-Back. Y estados Zero, Negative).
- 7x Resistencias de 220Ω a 330Ω
- 1x Display de 7 Segmentos (de Cátodo Común).
- 1x Cable para protoboard o UTP
- 5x Jumpers hembra-macho
- 1x Cable USB-C (para alimentación y programación).

B. Entorno de Desarrollo

- **Lenguaje de Programación: MicroPython**
 - **IDE Recomendado: Thonny**
 - **Configuración de Pines:** Todos los pines de botón deben usar configuración **Pull-Up** interna.
-

III. Estructura y Documentación del Repositorio

Cada pareja deberá crear un repositorio en **GitHub** con el nombre Micro-CPU-2Bits y organizar el contenido de la siguiente manera. La calidad de la documentación es un criterio clave de evaluación.



- Revisar detalle de cada sección en la Descripción de Secciones.
-

IV. Criterios de Evaluación

La evaluación del proyecto se centrará en los siguientes puntos:

1. **Funcionalidad (50%):**

- Implementación correcta del **Ciclo de Instrucción** (secuencia de LEDs).
- Detección correcta de **Banderas** (Zero, Negative, Overflow en el LED integrado).

2. **Código (30%):**

- Uso de **MicroPython** con la lógica de Pull-Up para botones.
- Código bien estructurado y documentado (uso de comentarios y funciones).

3. **Documentación (20%):**

- Cumplimiento de la **estructura de repositorio** solicitada.
- Claridad y precisión de los archivos README.md, PINOUT_FINAL.md y COMPONENTES.md.

Descripción de las Secciones del Repositorio

1. README.md (La Tarjeta de Presentación)

Este es el archivo más importante. Debe ser lo primero que vean y debe incluir:

- **Título del Proyecto:** Micro-CPU 2Bits
- **Autores:** Nombres y ID de los miembros de la pareja.
- **Objetivo:** Explicar brevemente qué simula el proyecto (Ciclo de Instrucción, Banderas Zero/Negative/Overflow).
- **Instrucciones de Uso:** Cómo cargar el código en la RP2040 Zero (usando Thonny) y cómo operar la calculadora (por ejemplo: "Presiona un dígito, luego una operación, luego otro dígito y finalmente '=').

2. CODE/ (El Código Fuente)

Contiene todo el código necesario para que el proyecto funcione.

- main.py: El código fuente completo en **MicroPython** que desarrollamos, incluyendo la máquina de estados, el manejo de LEDs y el control del display.

3. DOCUMENTATION/ (La Guía del Hardware)

Archivos de texto o Markdown que detallan los aspectos técnicos de la construcción.

- PINOUT_FINAL.md: Una lista clara y concisa de **todos los pines GPIO** utilizados y a qué componente están conectados (por ejemplo: "GPIO7 -> LED FETCH", "GPIO4 -> Botón '+'"). Esto es vital para depurar.
- COMPONENTES.md: El **listado de elementos**, especificando la cantidad, el tipo (por ejemplo: "7x Resistencias de 220 Ohmios") y las especificaciones clave (por ejemplo: "Display de 7 Segmentos: Cátodo Común").

4. HARDWARE/ (Los Archivos de Diseño)

Contiene cualquier archivo generado por software de diseño.

- Diagrama_Circuito.png o .svg: La imagen del **diagrama de conexiones** final que crearon en Cirkuit Designer IDE o similar.

5. 📁 MEDIA/ (Las Evidencias) 📁

Contiene las pruebas visuales de que el proyecto funciona.

- Foto_Ensamblaje_01.jpg, Foto_Final_02.jpg, etc.: Varias fotografías del proyecto **ensamblado y funcionando** (Fotografías claras y bien enfocadas).
- Video_Funcionamiento.mp4 (Opcional, pero recomendado): Un clip corto donde se muestre el ciclo completo de una operación ($1 + 2 = 3$) y una operación con overflow ($2 + 2 = 4$).