

Medidor de distancia por ultrasonidos de alta precisión

Raspberry Pi Pico (MicroPython + PIO) · 2 × TCT40 · electrónica discreta del inventario

1. Objetivo y resultado

Diseñar un medidor de distancia por tiempo de vuelo con **dos transductores TCT40 separados** (uno emisor, uno receptor), gobernado por una Raspberry Pi Pico programada en MicroPython, usando exclusivamente componentes del inventario disponible, y **maximizando la precisión**.

El resultado del diseño, verificado bloque a bloque con simulación SPICE (PySpice + ngspice) y análisis Monte Carlo:

Magnitud	Valor obtenido
Precisión (1σ) a 1 m, con compensación térmica	≈ 0,63 mm
Dispersión debida a tolerancias de componentes	0,26 mm (con interpolación por cruce por cero)
Dispersión si se midiera solo con el comparador	9,0 mm
Zona ciega (ring-down medido)	≈ 64 mm
Alcance de diseño	≈ 4 m
Resolución del cronómetro (PIO)	16 ns ≈ 2,7 μm

La idea central del diseño es que **la precisión no la limita el reloj sino la detección**. Un RP2040 a 125 MHz cronometra con 8 ns de resolución, equivalente a 1,4 μm; sin embargo, un detector de umbral clásico dispara antes o después según la amplitud del eco (error de *walk*), lo que arruina la medida en varios milímetros. Todo el diseño se articula en torno a eliminar ese error.

2. Arquitectura

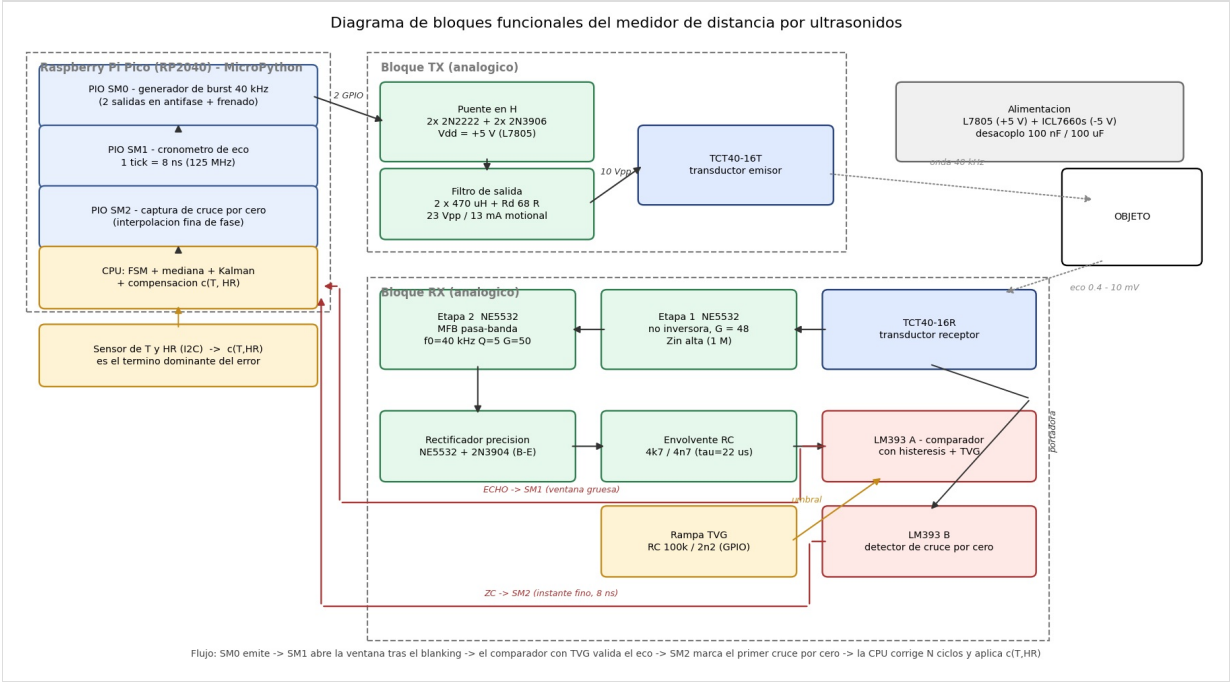


Diagrama de bloques funcionales

La cadena de medida trabaja en **dos escalas de tiempo simultáneas**:

- Escala gruesa (validación).** El comparador de envolvente confirma que ha llegado un eco real y no ruido. Su instante de disparo depende de la amplitud, así que sirve para *decidir*, no para *medir*.
- Escala fina (medida).** La fase de la portadora de 40 kHz es prácticamente independiente de la amplitud del eco. El primer cruce por cero tras la validación proporciona un instante estable con dispersión de ~1,5 μs. Esta es la medida

buena.

La CPU combina ambas: la gruesa resuelve la ambigüedad de N ciclos de 25 μ s que tiene la fina, y la fina aporta la precisión. Esta técnica es la misma que emplean los caudalímetros ultrasónicos comerciales.

Fuente Mermaid del diagrama

```
graph LR
    subgraph PICO ["Raspberry Pi Pico - RP2040"]
        SM0["PIO SM0 · burst 40 kHz · 2 salidas en antifase"]
        SM1["PIO SM1 · cronómetro de eco · 8 ns"]
        SM2["PIO SM2 · captura de cruce por cero"]
        CPU["CPU · FSM + mediana + Kalman + c(T,HR)"]
    end

    subgraph TX ["Bloque TX"]
        HB["Puente en H · 2N2222 x2 + 2N3906 x2"]
        LC["Filtro de salida · 2 x 470 μH + Rd 68 Ω"]
        XT["TCT40-16T"]
    end

    subgraph RX ["Bloque RX"]
        XR["TCT40-16R"]
        A1["Etapa 1 · NE5532 no inversora G=48"]
        A2["Etapa 2 · NE5532 MFB pasa-banda f0=40 kHz Q=5 G=50"]
        RECT["Rectificador de precisión · NE5532 + 2N3904"]
        ENV["Envolvente RC · 4k7 / 4n7"]
        CMP["LM393 A · comparador con TVG"]
        ZC["LM393 B · cruce por cero"]
    end

    TVG["Rampa TVG · RC 100k/2n2 desde GPIO"]
    TEMP["Sensor T/HR · I2C"]
    OBJ(("Objeto"))

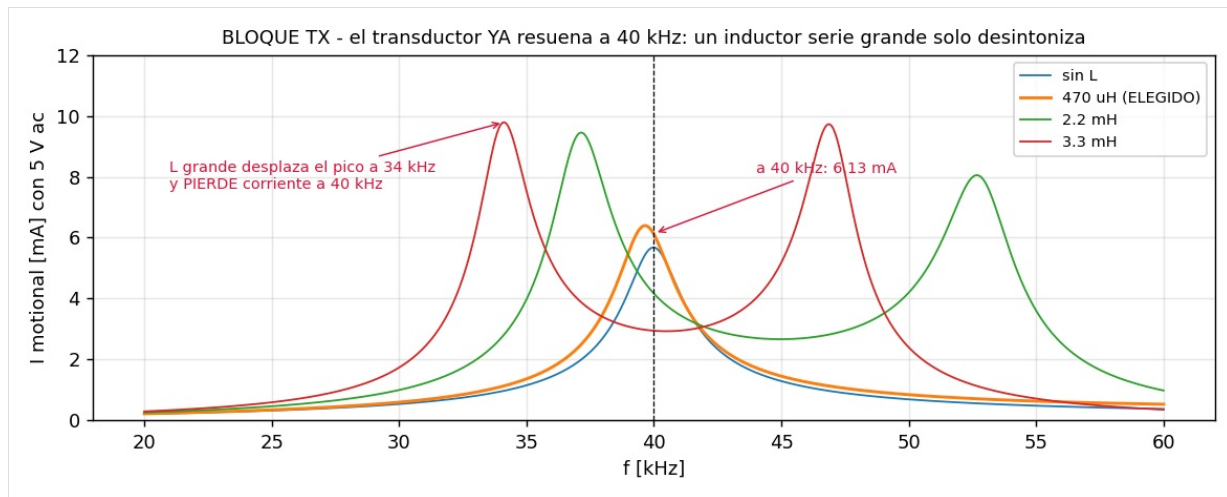
    SM0 --> HB --> LC --> XT
    XT -->|onda 40 kHz| OBJ
    OBJ -->|eco 0,4 a 10 mV| XR
    XR --> A1 --> A2 --> RECT --> ENV --> CMP --> SM1
    A2 --> ZC --> SM2
    TVG -->|umbral| CMP
    SM1 -->|irq 5| SM2
    SM1 --> CPU
    SM2 --> CPU
    TEMP --> CPU
    CPU --> OUT["UART / I2C"]
```

3. Bloque TX — excitación del emisor

3.1 Un resultado contraintuitivo de la simulación

El TCT40 se comporta eléctricamente como un modelo Butterworth–Van Dyke: una capacidad estática **$C_0 \approx 2,4 \text{ nF}$** en paralelo con una rama mecánica serie R_m – L_m – C_m resonante a 40 kHz. Los GPIO del Pico solo dan 3,3 V, así que la primera medida obvia es un **punto en H** ($2 \times 2N2222 \text{ NPN} + 2 \times 2N3906 \text{ PNP}$) alimentado a 5 V desde el L7805, que lleva la excitación a 10 Vpp diferenciales.

La segunda idea, aparentemente natural, era añadir un inductor en serie para resonar con C_0 y multiplicar la tensión. **La simulación la desmonta.** El error está en optimizar la magnitud equivocada: lo que produce sonido no es la tensión en bornes, sino la **corriente por la rama mecánica**. Y a 40 kHz el transductor *ya está* en resonancia serie, es decir, ya presenta su impedancia mínima. Añadir inductancia en serie no amplifica: desplaza el punto de resonancia del conjunto y reduce la corriente justo donde la queremos.



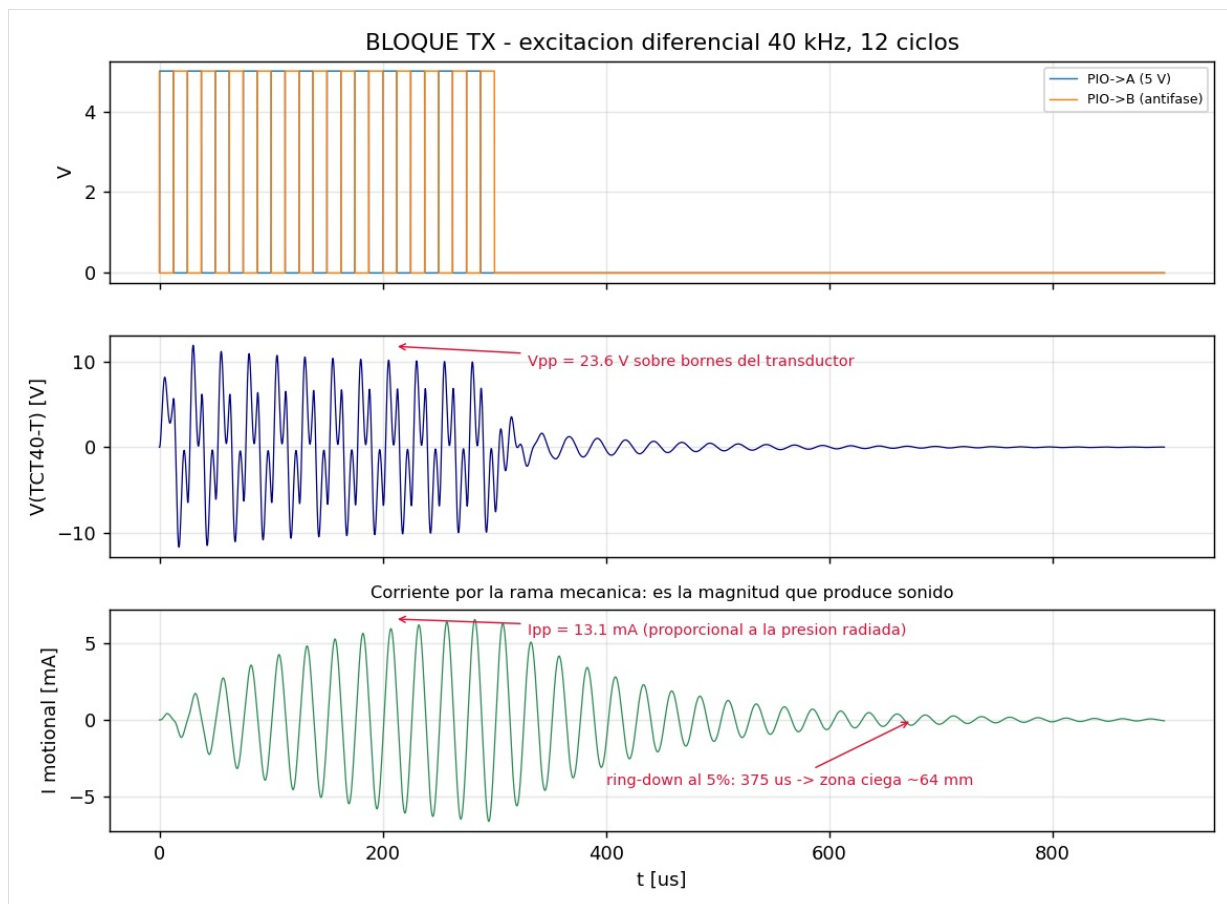
Elección del inductor serie

Corriente mecánica a 40 kHz con 5 V de excitación, según el barrido AC:

Inductor serie por rama	Corriente a 40 kHz
Sin inductor	5,68 mA
470 μH (elegido)	6,13 mA
2,2 mH	4,18 mA
3,3 mH	2,94 mA

Con 3,3 mH el pico se va a 34 kHz y se pierde la mitad de la señal. El óptimo está en un valor pequeño: **470 μ H por rama**, disponible en el kit Bojack, que aporta un 8 % de mejora y además filtra los armónicos de la onda cuadrada. Se añade **Rd = 68 Ω** en serie con cada rama para limitar la corriente de pico del puente.

3.2 Simulación transitoria



Simulación TX: burst y ring-down

Medidas obtenidas de la simulación con 12 ciclos de excitación:

- **23,6 Vpp** en bornes del transductor (superior a los 10 Vpp del puente porque la corriente mecánica circula por C_0).
- **13,1 mA_{pp}** de corriente mecánica, que es la magnitud proporcional a la presión radiada.
- **Ring-down hasta el 5 % = 375 μ s**, lo que impone una **zona ciega de ~64 mm**. Este tiempo lo fija el factor de calidad mecánico del propio transductor ($Q_m \approx 20$), no el circuito: es una limitación física del TCT40 que no se puede rodear con electrónica.

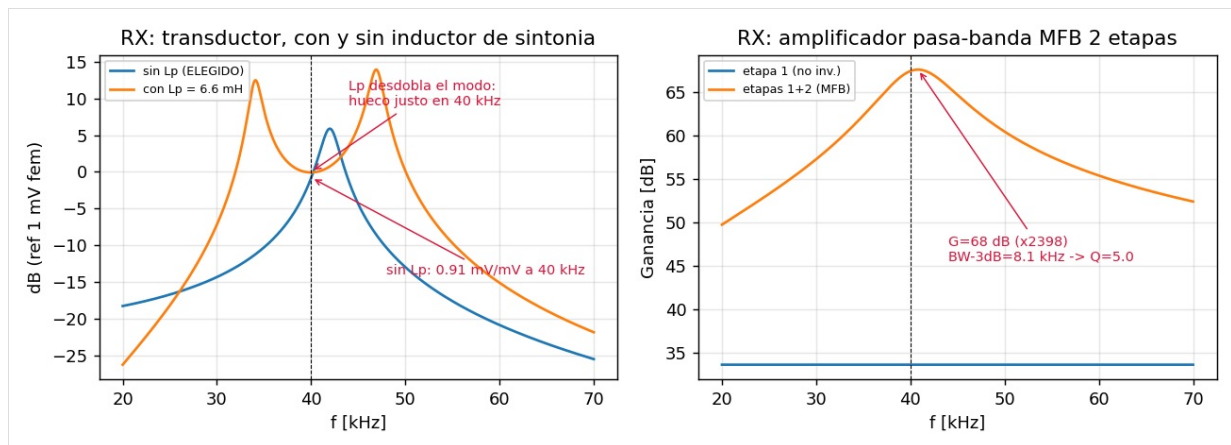
Se probó **frenado activo** (excitar 1 a 4 ciclos en antifase al final del burst) y el óptimo apenas reduce el ring-down un 10 %. La conclusión honesta de la simulación es que **no compensa** salvo que se enganche en fase con la oscilación real; el firmware deja el mecanismo preparado pero por defecto frena por cortocircuito de lado bajo (ambas salidas a nivel bajo).

El compromiso del número de ciclos es directo: más ciclos dan más energía y más alcance, pero alargan el burst y la envolvente. Doce ciclos (300 μ s) es un buen punto medio; la precisión no se resiente porque no la determina la envolvente sino el cruce por cero.

4. Bloque RX — transductor y amplificación

4.1 Decisión de diseño: sin inductor de sintonía

Es habitual poner un inductor en paralelo con el transductor receptor para resonar con C_0 y obtener ganancia de tensión gratis. **La simulación desaconseja hacerlo aquí.**



Simulación RX: respuesta AC

Con el modelo BVD completo (rama mecánica resonando también a 40 kHz), el conjunto se comporta como dos resonadores acoplados: el modo se desdobra en **34 kHz y 47 kHz, dejando un hueco justo en 40 kHz**, que es exactamente donde queremos ganancia. Un barrido de L_p entre 3,3 mH y 33 mH confirmó que ninguna elección mejora la respuesta a 40 kHz. Se elimina el componente: **menos BOM, mejor respuesta y una fuente de tolerancia menos**.

4.2 Cadena de amplificación

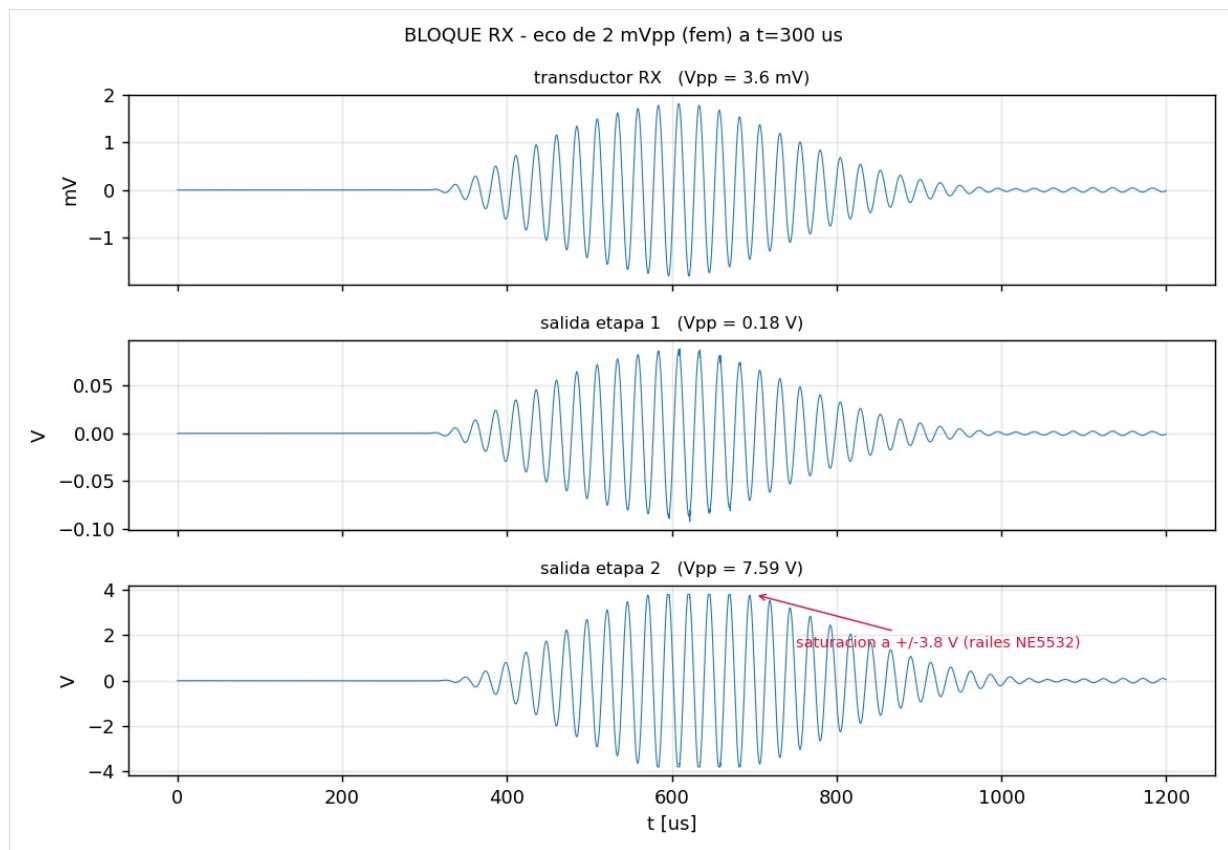
La ganancia necesaria es alta: el eco útil va de ~10 mV (objeto cercano) a ~0,4 mV (objeto a más de un metro), y el comparador necesita voltios.

- **Etapas 1 — NE5532 en configuración no inversora, $G = 48$.** Se elige no inversora precisamente por su impedancia de entrada altísima: una etapa inversora presentaría sus ~390 Ω de resistencia de entrada al transductor y hundiría la señal. La realimentación lleva un condensador de 470 nF en serie con la resistencia de ganancia, de modo que la **ganancia en continua es 1** y el offset del amplificador no se multiplica.
- **Etapas 2 — NE5532 en filtro pasa-banda MFB, $f_0 = 40$ kHz, $Q = 5$, $G = 50$,** con $R = 390 \Omega / 39$ k Ω y $C = 1$ nF (valores presentes en los kits).

Se usa NE5532 y no LM358 por una razón cuantitativa: a 40 kHz el LM358 (GBW 1 MHz) solo tiene 25 de ganancia en lazo abierto, insuficiente para cerrar un lazo de 50 con precisión de fase; el NE5532 (GBW 10 MHz) ofrece 250. **La estabilidad de fase importa aquí más que la ganancia**, porque el instante de cruce por cero es la medida.

Respuesta medida en simulación: **68 dB ($\times 2400$) de ganancia total, BW-3 dB = 8,1 kHz, $Q \approx 5$.** El filtrado es esencial: reduce el ruido de banda ancha del amplificador y rechaza interferencias fuera de los 40 kHz.

4.3 Transitorio



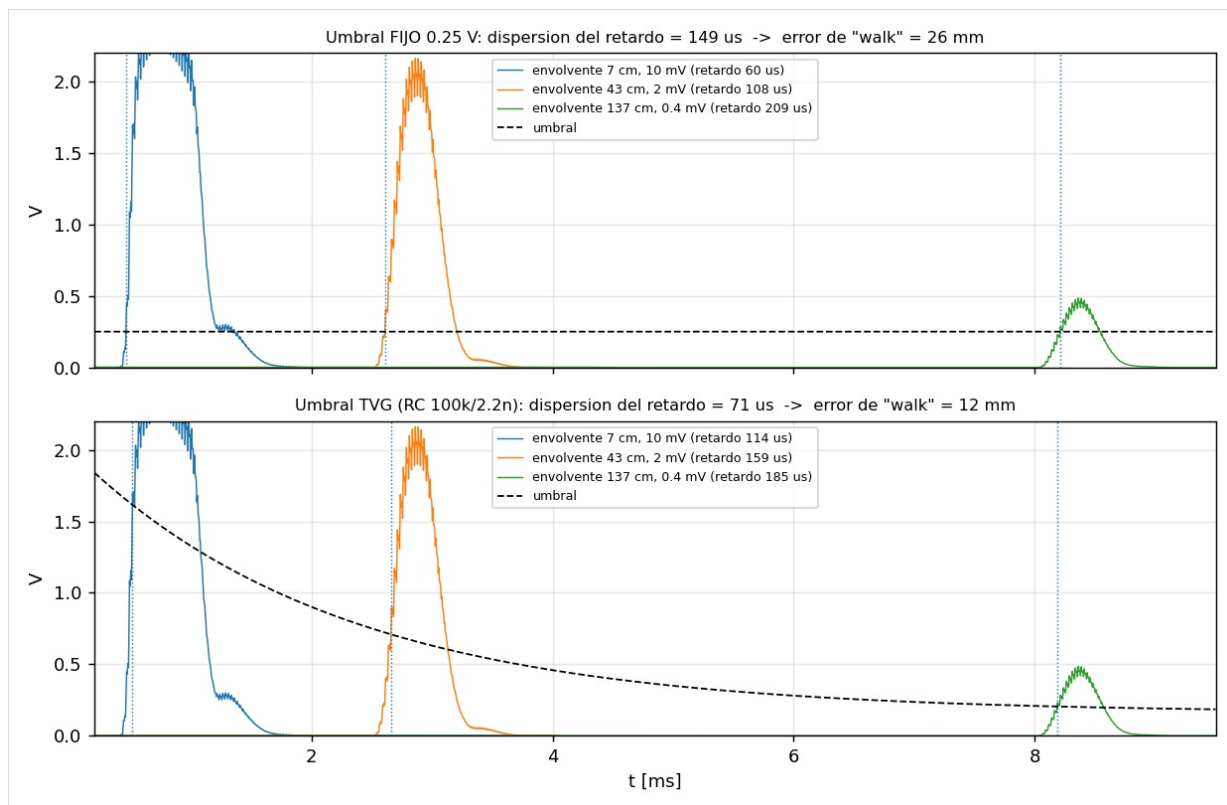
Con un eco de 2 mVpp, la etapa 1 entrega 0,18 Vpp y la etapa 2 satura contra los raíles ($\pm 3,8$ V con alimentación ± 5 V). **La saturación es aceptable y hasta deseable:** no distorsiona los cruces por cero, que es lo que medimos, y garantiza que el comparador de fase tenga siempre flancos limpios. La alimentación simétrica se obtiene con L7805 y el convertidor de carga ICL7660s del kit de circuitos integrados.

5. Bloque de detección — el corazón de la precisión

5.1 El problema del *walk*

El eco se rectifica con un **rectificador de precisión** (NE5532 con la unión base-emisor de un 2N3904 como diodo, ya que **el inventario no incluye diodos**; el lazo de realimentación del amplificador cancela la caída de 0,6 V) y se integra con una red RC de 4k7 / 4n7 ($\tau = 22 \mu\text{s}$).

Un comparador con umbral fijo dispara cuando la envolvente lo cruza, y ese instante llega tarde y de forma **dependiente de la amplitud**: un eco fuerte cruza pronto, uno débil tarde.



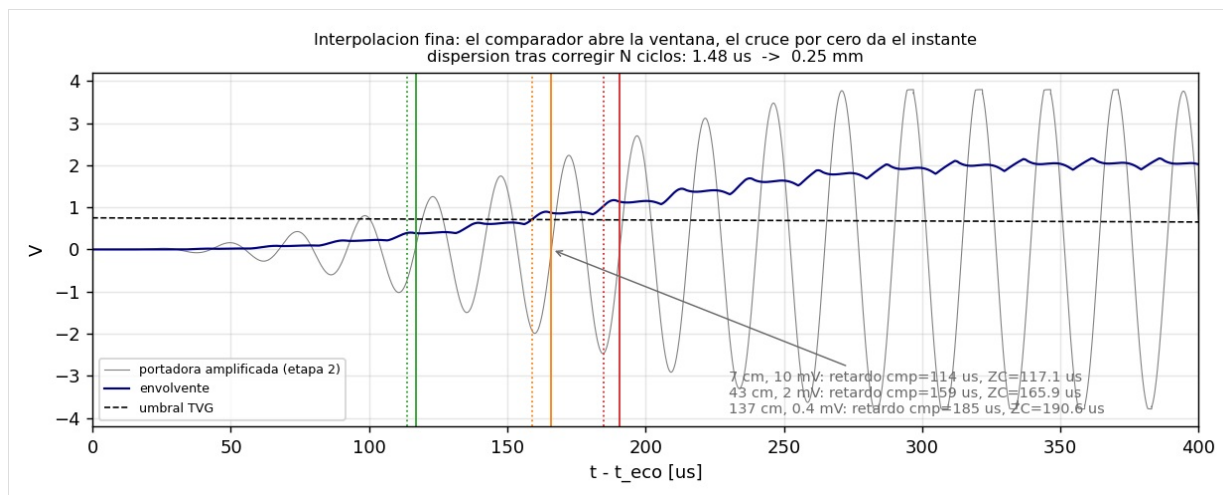
Simulando tres ecos con amplitudes realistas para su distancia (10 mV a 7 cm, 2 mV a 43 cm, 0,4 mV a 137 cm):

Estrategia	Dispersión del retardo	Error de distancia
Umbral fijo 0,25 V	149 μ s	26 mm
Umbral TVG (rampa RC 100 k/2n2)	71 μ s	12 mm

El **TVG** (*Time Varied Gain*, aquí implementado como umbral variable en el tiempo) es una rampa exponencial decreciente generada con una red RC que un GPIO carga al inicio del burst y luego deja libre poniéndose en alta impedancia. Como la amplitud del eco decae con la distancia, un umbral que decae en paralelo mantiene aproximadamente constante la relación umbral/amplitud. Reduce el walk a la mitad y, de propina, **rechaza el ring-down** del propio emisor.

Pero 12 mm sigue siendo un error inaceptable.

5.2 La solución: interpolación por cruce por cero



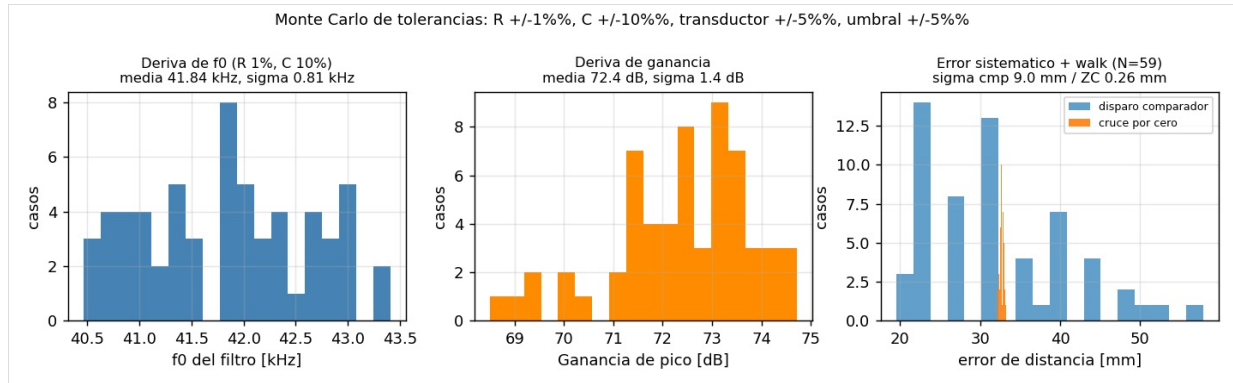
El segundo comparador del LM393 vigila la **portadora amplificada**, no la envolvente. Sus cruces por cero ocurren en instantes fijados por la fase de la onda acústica, que **no depende de la amplitud del eco**.

El procedimiento es:

1. El comparador de envolvente con TVG abre una ventana: “aquí hay un eco de verdad”.
2. Se toma el **primer cruce por cero posterior** como instante de llegada.
3. Como los cruces se repiten cada 25 μs , hay una ambigüedad de N ciclos que se resuelve con la medida gruesa (cuya incertidumbre, 12 μs , es menor que medio ciclo).

Resultado simulado: la dispersión cae de 71 μs a **1,48 μs , es decir 0,25 mm**. Una mejora de casi $\times 50$ sin añadir un solo componente caro.

6. Análisis de tolerancias (Monte Carlo)



Monte Carlo de tolerancias

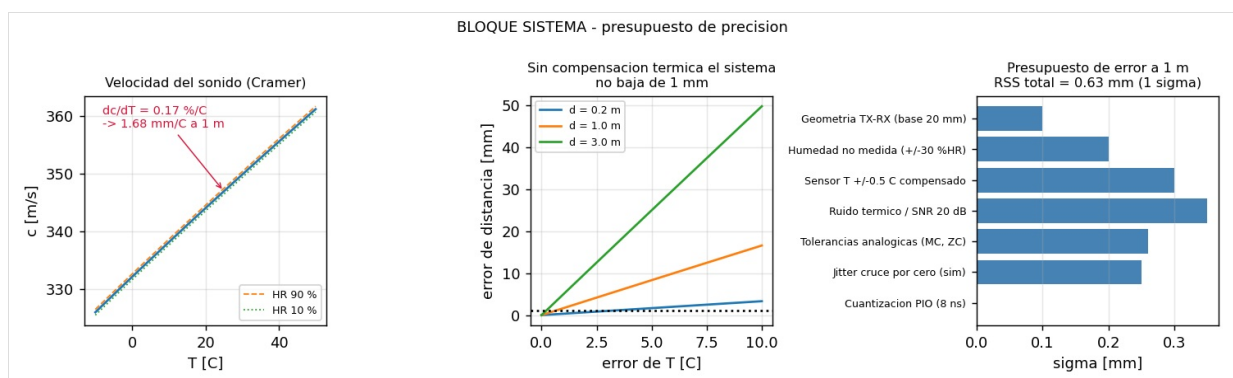
Se ejecutaron 80 simulaciones transitorias y 60 barridos AC con valores aleatorios dentro de tolerancias realistas: **resistencias $\pm 1\%$, condensadores $\pm 10\%$, parámetros del transductor $\pm 5\%$, umbral $\pm 5\%$** , y amplitud de eco variando en un rango de 28 dB.

Parámetro	Resultado
f_0 del filtro pasa-banda	$41,84 \pm 0,81$ kHz
Ganancia de pico	$72,4 \pm 1,4$ dB
Error de distancia con comparador	$\sigma = 9,0$ mm
Error de distancia con cruce por cero	$\sigma = 0,26$ mm

Tres conclusiones prácticas:

1. **El condensador domina la deriva de frecuencia.** Con C al $\pm 10\%$, f_0 se dispersa $\pm 0,8$ kHz. Como el filtro tiene 8,1 kHz de ancho de banda, es tolerable, pero conviene **usar condensadores de 1 nF de buena calidad (NP0/COG) y medirlos**, o ajustar con la serie E24 disponible.
2. **La media de f_0 sale desplazada a 41,8 kHz**, no a 40,0 kHz. El desplazamiento proviene de la interacción del filtro MFB con el polo del amplificador y de la asimetría de las tolerancias. En el diseño real conviene **recentrar el valor nominal de R3 a ≈ 43 k Ω** o verificar con un barrido antes de fijar el BOM.
3. **La interpolación por cruce por cero hace el sistema casi inmune a las tolerancias:** factor 35 de mejora. Las tolerancias afectan a la *ganancia* y al *retardo de grupo*, pero un retardo de grupo constante es un simple offset de calibración, no un error aleatorio.

7. Presupuesto de error y límite físico



Aquí aparece el hecho incómodo: **una vez resuelto el problema electrónico, el limitante ya no es el circuito sino el aire.**

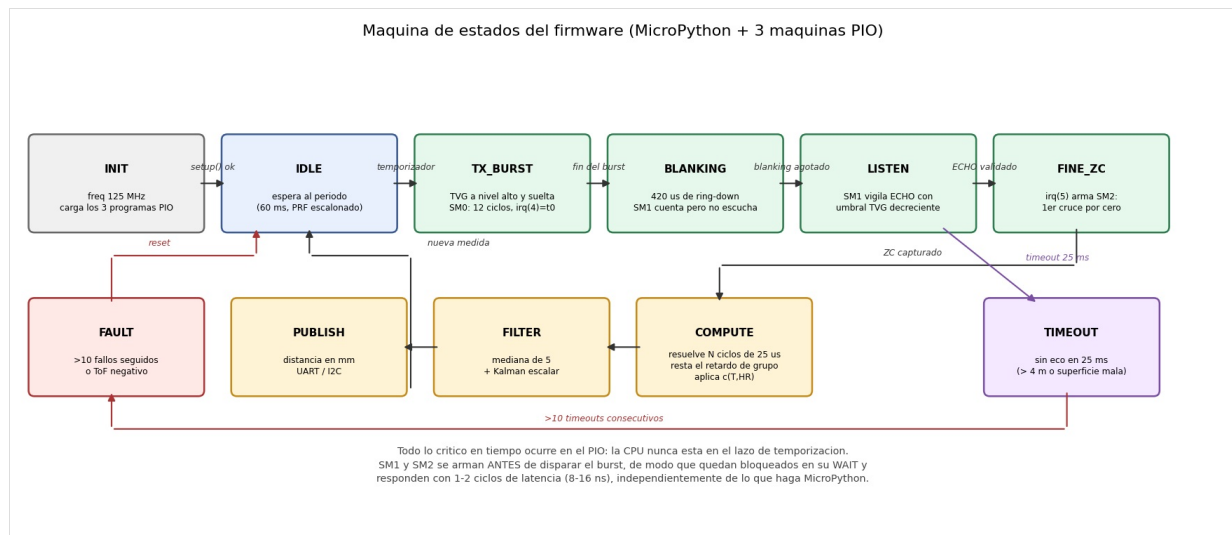
La velocidad del sonido depende de la temperatura según $\approx 0,17 \text{ }^\circ\text{C}$, lo que a 1 m supone **1,7 mm por cada $^\circ\text{C}$ de error**. Sin compensación térmica el sistema no puede bajar del milímetro por mucho que se refine el analógico. La humedad influye mucho menos ($\approx 0,2 \text{ mm}$ a 1 m para un rango completo de humedad relativa) y la presión es despreciable.

Por eso el diseño incluye **un sensor de temperatura y humedad por I2C** como parte integral del instrumento, no como accesorio. Con un sensor de $\pm 0,5 \text{ }^\circ\text{C}$:

Contribución	$\sigma \text{ (mm) a 1 m}$
Cuantización del PIO (8 ns)	0,001
Jitter del cruce por cero (simulado)	0,25
Tolerancias analógicas (Monte Carlo)	0,26
Ruido térmico / SNR 20 dB	0,35
Sensor de temperatura $\pm 0,5 \text{ }^\circ\text{C}$ compensado	0,30
Humedad no medida ($\pm 30 \text{ } \%$ HR)	0,20
Geometría TX-RX (base de 20 mm)	0,10
Total RSS (1σ)	0,63 mm

Nota sobre la geometría: con transductores separados, el camino es un triángulo, no una recta. Con una base b entre centros, la corrección es $d = \sqrt{((c \cdot t/2)^2 - (b/2)^2)}$. Con $b = 20 \text{ mm}$ el error a 1 m es de solo $0,05 \text{ mm}$, pero **a 10 cm sería de $0,5 \text{ mm}$** , así que el firmware debe aplicar la corrección. Conviene montar los transductores lo más juntos posible y con un tabique acústico entre ellos para evitar la diafonía directa.

8. Software: máquina de estados y uso del PIO



Máquina de estados

8.1 Por qué PIO y no interrupciones

MicroPython tiene una latencia de interrupción de decenas de microsegundos, con jitter: mediría con un error de centímetros. El **PIO del RP2040** ejecuta programas determinísticos con precisión de un ciclo de reloj (8 ns) sin intervención de la CPU. Se usan **tres máquinas de estado**:

SM	Función	Detalle
SM0	Genera el burst de 40 kHz	Reloj a 2 MHz (divisor 62,5, exacto , sin error de frecuencia). 50 ciclos PIO por periodo: 25 en una fase y 24 + 1 del salto en la otra. Emite <code>irq(4)</code> como marca to.
SM1	Cronómetro grueso	Espera <code>irq(4)</code> , consume el blanking programable, y luego cuenta hacia atrás vigilando el pin ECHO. Bucle de 2 ciclos → resolución 16 ns . Al detectar emite <code>irq(5)</code> .
SM2	Cronómetro fino	Bloqueado en <code>wait(1, irq, 5)</code> ; al dispararse mide el retardo hasta el primer cruce por cero. Latencia de armado: 1-2 ciclos.

El detalle clave de implementación: **SM1 y SM2 se arman antes de disparar el burst**, de modo que quedan bloqueados en su instrucción `wait` y responden inmediatamente. La CPU nunca está en el lazo de temporización.

Las dos escalas se suman: $t_{\text{vuelo}} = t_{\text{blanking}} + (X_0 - x_1) \cdot 2 \cdot 8 \text{ ns} + (X_0 - x_2) \cdot 2 \cdot 8 \text{ ns}$, y después se corrige la ambigüedad de N ciclos.

8.2 Estados

Estado	Qué hace
INIT	Fija el reloj a 125 MHz y carga los tres programas PIO
IDLE	Espera el periodo de repetición (60 ms, con escalonado aleatorio del PRF para rechazar ecos de la medida anterior y otros sensores)
TX_BURST	Carga la rampa TVG por GPIO, arma SM1/SM2 y lanza los 12 ciclos
BLANKING	420 μs durante los que se ignora la entrada (ring-down medido: 375 μs + margen)
LISTEN	SM1 vigila el comparador con umbral TVG decreciente
FINE_ZC	SM2 captura el primer cruce por cero
COMPUTE	Resuelve N ciclos, resta el retardo de grupo, aplica $c(T, HR)$ y la corrección geométrica
FILTER	Mediana de 5 (elimina ecos falsos sin retardar escalones) + Kalman escalar (aliso fino). El Kalman ajusta su varianza de medida según se haya conseguido o no el cruce por cero
PUBLISH	Entrega la distancia
TIMEOUT	Sin eco en 25 ms
FAULT	Más de 10 fallos consecutivos o tiempo de vuelo negativo

8.3 Calibración

El único parámetro a calibrar es el **retardo de grupo de la cadena analógica** ($\approx 166 \mu\text{s}$ según simulación), que es un offset constante. Se determina midiendo un objeto a distancia conocida. Todo lo demás es determinista o se compensa en tiempo real.

9. Lista de materiales (BOM)

Esta lista se genera automáticamente desde la misma fuente que el netlist de KiCad, de modo que las referencias coinciden con el esquemático. Todo procede del inventario aportado salvo el Pico, los dos TCT40 y el sensor de temperatura/humedad.

Corrección respecto al diseño conceptual: el puente en H necesita un **pre-driver NPN por rama** (Q1 y Q4). Un GPIO de 3,3 V no puede cortar un PNP cuyo emisor está a 5 V, porque quedaría una V_{be} de 1,7 V y el transistor conduciría permanentemente. El pre-driver invierte y desplaza el nivel; añade unos 100 ns de retardo idéntico en ambas ramas, así que no introduce error de temporización.

También se añadieron el atenuador del comparador de cruce por cero (R26/R27) y la referencia VREF (R28/R29/C7): el LM393 se alimenta con +5 V/GND y su rango de modo común no admite las excursiones negativas de la portadora, así que

ésta se acopla en alterna y se centra en 2,5 V. El instante del cruce no se altera, porque la atenuación es lineal y la referencia del comparador es la misma VREF.

Alimentacion

Ref.	Valor	Función
J3	Conn_01x02	Entrada de alimentacion 7-12 V DC
U4	L7805	Regulador +5 V
U5	ICL7660S	Convertidor de carga a -5 V
C8	470u/25V	Reservorio de entrada
C9	220u/25V	Reservorio +5 V
C10	10u/25V	Bomba de carga del ICL7660
C11	10u/25V	Salida del ICL7660
R32	10	Filtro del rail -5 V (con C12)
C12	100u/25V	Filtro del rail -5 V
C19	470u/16V	Reservorio local del puente en H

Interfaz Pico / sensor

Ref.	Valor	Función
J1	Conn_01x10	GP2,GP3,GP4,GP5,GP6,3V3,GND,GND,GP8_SDA,GP9_SCL
J2	Conn_01x04	Sensor T/HR I2C (3V3, GND, SDA, SCL)
R30	4k7	Pull-up SDA
R31	4k7	Pull-up SCL

TX

Ref.	Valor	Función
R1	1k	Base del pre-driver A
Q1	2N2222	Pre-driver A
R2	2k2	Pull-up del pre-driver A
R3	470	Base PNP lado alto A
Q2	2N3906	Lado alto A
R4	1k	Base NPN lado bajo A
Q3	2N2222	Lado bajo A
R5	1k	Base del pre-driver B
Q4	2N2222	Pre-driver B
R6	2k2	Pull-up del pre-driver B
R7	470	Base PNP lado alto B
Q5	2N3906	Lado alto B
R8	1k	Base NPN lado bajo B
Q6	2N2222	Lado bajo B
R9	68	Amortiguacion rama A
L1	470u	Filtro rama A
R10	68	Amortiguacion rama B
L2	470u	Filtro rama B
LS1	TCT40-16T	Transductor emisor
C13	100n	Desacoplo +5 V del puente

RX - amplificacion

Ref.	Valor	Función
MIC1	TCT40-16R	Transductor receptor
R11	1M	Polarizacion de entrada
U1	NE5532	Etapas 1 (A) y filtro MFB (B)
R12	47k	Realimentacion etapa 1
R13	1k	Ganancia etapa 1 ($G = 1 + R12/R13 = 48$)
C1	470n	Ganancia unitaria en continua
R14	390	Entrada del filtro MFB
C2	1n NP0	Filtro MFB
C3	1n NP0	Filtro MFB
R15	39k	Realimentacion MFB ($f_0 = 40 \text{ kHz}$, $Q = 5$, $G = 50$)
C14	100n	Desacoplo U1 +5 V
C15	100n	Desacoplo U1 -5 V

RX - rectificador y envolvente

Ref.	Valor	Función
U2	NE5532	Rectificador de precision (A); B sin usar
R16	10k	Entrada del rectificador
Q7	2N3904	Union B-E usada como diodo (no hay diodos en el kit)
R17	10k	Realimentacion del rectificador
R18	4k7	Envolvente
C4	4n7	Envolvente ($\tau = 22 \text{ us}$)
R19	47k	Descarga de la envolvente
C16	100n	Desacoplo U2 +5 V
C17	100n	Desacoplo U2 -5 V

Deteccion (comparadores y TVG)

Ref.	Valor	Función
U3	LM393	A: envolvente vs TVG. B: cruce por cero
R20	1M	Histeresis del comparador de envolvente
R21	4k7	Pull-up ECHO a 3V3 (colector abierto)
R22	4k7	Pull-up ZC a 3V3 (colector abierto)
R23	220	Limitacion de corriente del GPIO de TVG
R24	3M	Suelo del umbral TVG ($\sim 0,22 \text{ V}$)
R25	220k	Descarga TVG
C5	10n	TVG: $\tau = (3M)$
C6	100n	Acoplo del cruce por cero
R26	33k	Atenuador del cruce por cero
R27	10k	Atenuador del cruce por cero
R28	100k	Divisor de VREF
R29	100k	Divisor de VREF
C7	100n	Desacoplo de VREF
C18	100n	Desacoplo U3

Nota sobre el ICL7660s: su bomba de carga conmuta a unos 10 kHz y puede inyectar ruido en la banda de interés. Por eso el rail negativo lleva el filtro R32/C12. Si el ruido resultara problemático, la alternativa es alimentar el analógico con dos pilas.

Componentes del inventario deliberadamente no usados: el LM358 (ancho de banda insuficiente a 40 kHz), el NE555 (el PIO genera los 40 kHz con precisión mucho mayor) y el inductor de sintonía del receptor (razonado en §4.1).

10. Recomendaciones de montaje

1. **Separación acústica.** Un tabique de espuma entre emisor y receptor evita el acoplamiento directo, que satura la entrada y alarga la zona ciega efectiva.
2. **Masas separadas.** La corriente del puente en H (picos de decenas de mA a 40 kHz) no debe compartir camino de retorno con la etapa 1, que maneja microvoltios. Un solo punto de unión de masas junto al condensador de reservorio.
3. **Guardas.** La pista de entrada del NE5532 debe ser corta y rodeada de masa; con 1 MΩ de polarización, cualquier fuga la afecta.
4. **Condensadores NP0/C0G** en el filtro MFB. Los cerámicos de alta constante (Y5V/Z5U) varían más del 10 % con la temperatura, y eso desplaza f_0 .
5. **No sustituir los 470 μH por un valor mayor** “para ganar tensión”: como muestra §3.1, la señal acústica empeora.

11. Limitaciones del análisis

Por honestidad técnica conviene explicitar qué no cubre este estudio:

- El **modelo BVD del TCT40** se ha construido con valores típicos ($C_0 = 2,4$ nF, $C_m = 250$ pF, $L_m = 63,4$ mH, $R_m = 800$ Ω, $Q_m \approx 20$), no medidos sobre las unidades reales. Los valores exactos afectan a la ganancia del tanque TX y al ring-down.
- Los amplificadores se han simulado con un **macromodelo de un polo** con saturación de raíles. Reproduce bien ganancia, ancho de banda y recorte, pero no el slew rate, el ruido ni la distorsión.
- La **propagación acústica** (divergencia del haz, absorción, reflexión en la superficie) se ha modelado imponiendo amplitudes de eco realistas, no resolviendo la física.
- El **ruido** solo se contempla en el presupuesto de error como un término estimado (SNR 20 dB); no se ha hecho un análisis de ruido SPICE de la cadena.

Ninguna de estas limitaciones afecta a la conclusión principal, que es de naturaleza estructural: **medir el instante por cruce por cero en lugar de por umbral mejora la precisión en un factor cercano a 35**, y a partir de ahí el límite lo pone la compensación de la velocidad del sonido.

12. Archivos entregados

Archivo	Contenido
medidor_ultrasonidos.pdf	Este documento
ultrasonido_pico.py	Firmware MicroPython con los tres programas PIO y la máquina de estados
simulaciones/	Scripts PySpice que generan todas las figuras
medidor_ultrasonidos.net	Netlist en formato KiCad (para importar en Pcbnew)
medidor_ultrasonidos.cir	Netlist SPICE del circuito completo, simulable
bom.csv	Lista de materiales en CSV
gen_net.py	Generador y verificador de ambos netlists