

# Theremin Digital cu ESP32

## Introducere

Proiectul constă în realizarea unui instrument muzical fără contact fizic, inspirat din thereminul clasic inventat de Lev Termen în 1920. Utilizatorul controlează frecvența sunetului prin mișcarea mâinii deasupra unui senzor ultrasonic — cu cât mâna este mai aproape, cu atât nota muzicală este mai înaltă.

- **Ce face:** Generează note muzicale în timp real în funcție de distanța mâinii față de senzor, afișând nota curentă, frecvența și octava pe un display OLED.
- **Scopul:** Demonstrarea integrării mai multor periferice hardware (ADC, PWM, I2C, întreruperi, timere) într-un sistem embedded interactiv și funcțional.
- **Ideea de pornire:** Thereminul clasic este unul dintre primele instrumente muzicale electronice, controlat prin gesturi fără contact. Am dorit să recreez acest concept folosind componente accesibile și un microcontroller modern (ESP32).
- **Utilitate:** Proiectul este un instrument didactic excelent pentru înțelegerea interacțiunii între senzori, procesare digitală a semnalului și generare audio PWM. De asemenea, poate fi folosit ca instrument muzical recreativ.

## Descriere generală



## Schema Electrica



Sistemul este construit în jurul microcontrollerului **ESP32 DevKit V1** și funcționează astfel:

1. **Senzorul ultrasonic HC-SR04** măsoară distanța față de mâna utilizatorului (2–400 cm). ESP32 trimite un puls de 10μs pe pinul TRIGGER, apoi măsoară durata pulsului primit pe ECHO folosind un timer intern. Deoarece HC-SR04 funcționează la 5V, iar GPIO-urile ESP32 sunt de 3.3V, pinul ECHO este conectat printr-un **divizor rezistiv** (10kΩ + 20kΩ) pentru a aduce semnalul la un nivel sigur.
2. **Distanța este mapată la o frecvență muzicală** corespunzătoare unei note din gama cromatică (Do, Re, Mi, Fa, Sol, La, Si). Frecvența rezultată este aplicată ca semnal **PWM** pe un **buzzer pasiv piezoelectric** (CPT-1775-3TH), care produce tonul audibil.
3. Un **potențiomtru de 10kΩ** conectat la un canal **ADC** permite utilizatorului să selecteze octava activă (ex: octava 3 = 131–262 Hz, octava 5 = 523–1047 Hz), scalând întregul interval de frecvențe.

- Un **buton tactil** conectat la un pin de **întrerupere externă** permite activarea/dezactivarea sunetului (funcția mute). Apăsarea butonului comută instantaneu starea, fără a depinde de bucla principală.
- Display-ul OLED SSD1306** (128×64 pixeli), interfațat prin **I2C**, afișează în timp real: nota muzicală curentă (ex: „La”), frecvența exactă (ex: „440 Hz”) și octava selectată.
- Prin **UART**, ESP32 transmite date de debug către un terminal PC (nota, frecvența, distanța, starea mute).

Fluxul de date: HC-SR04 → Timer (măsurare ecou) → Calcul distanță → Mapare frecvență × octavă (ADC) → PWM buzzer + I2C OLED + UART debug.

## Hardware Design

### Lista de piese

Nr.	Componentă	Model / Specificații	Rol în circuit	Datasheet / Link
1	Microcontroller	ESP32 DevKit V1 (dual-core, WiFi, BT)	Unitate centrală de procesare	<a href="#">Espressif ESP32</a>
2	Senzor ultrasonic	HC-SR04 (5V, 2-400cm, rezoluție 3mm)	Măsurare distanță mână-senzor	<a href="#">HC-SR04 Datasheet</a>
3	Buzzer pasiv piezoelectric	CPT-1775-3TH (17mm, 3Vp-p, 75dB, Through Hole)	Generare ton muzical prin PWM	<a href="#">Same Sky CPT Series</a>
4	Display OLED	SSD1306, 0.96”, 128×64, I2C, 3-5V, alb	Afișare notă, frecvență, octavă	<a href="#">SSD1306 Datasheet</a>
5	Potențiometru	10kΩ, through hole	Selectare octavă activă (ADC)	-
6	Buton tactil	SPST, Through Hole (PC Mount)	Funcția mute (întrerupere externă)	-
7	Rezistență	10kΩ, Metal Film, 1%, Through Hole (RLR07C1002FS)	Divizor tensiune ECHO (sus)	-
8	Rezistență	20kΩ, Metal Film, 1%, Through Hole (RLR07C2002FS)	Divizor tensiune ECHO (jos, la GND)	-
9	Breadboard	830 puncte	Prototipare fără lipire	-
10	Fire conexiune	Jumper wires male-to-male	Conexiuni breadboard	-
11	Cablu USB	Micro-USB sau USB-C (depinde de varianta ESP32)	Alimentare + programare + UART	-

### Pinout ESP32

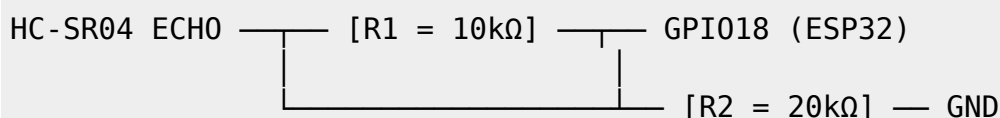
Periferic	Pin ESP32	Funcție
HC-SR04 TRIGGER	GPIO5	Output digital (puls 10μs)
HC-SR04 ECHO	GPIO18 (prin divizor rezistiv)	Input digital (măsurare durată puls)
Buzzer CPT-1775-3TH (+)	GPIO25	PWM output (frecvență variabilă)
Potențiometru (wiper)	GPIO34 (ADC1_CH6)	Input analogic (selectare octavă)

Buton mute	GPIO4	Înterupere externă (INPUT_PULLUP)
OLED SDA	GPIO21	I2C Data
OLED SCL	GPIO22	I2C Clock
HC-SR04 VCC	VIN (5V)	Alimentare senzor
OLED VCC	3.3V	Alimentare display

## Schema electrică

Schema electrică detaliată va fi adăugată ulterior.

### Divizor rezistiv pentru ECHO (5V → 3.3V):



$$V_{out} = 5V \times 20k / (10k + 20k) = 3.33V$$

## Software Design

- **Mediu de dezvoltare:** Arduino IDE 2.x cu board support ESP32 (Espressif Arduino Core)
- **Librării 3rd-party:**
  - **Adafruit SSD1306** — driver pentru display-ul OLED prin I2C
  - **Adafruit GFX** — funcții grafice (text, forme) pentru OLED
  - **NewPing** (opțional) — librărie pentru senzori ultrasonici (alternativă la implementare manuală cu pulseIn)

## Algoritmi și structuri planificate

- **Măsurare distanță:** Triggerul trimite puls de 10μs, apoi se măsoară durata ecou-ului. Distanța = (durată × 0.034) / 2 cm.
- **Mapare distanță → frecvență:** Intervalul util (ex: 2-30 cm) este mapat liniar la frecvențele notelor muzicale din octava selectată. Frecvențele notelor urmează scala temperată egală:  $f = 440 \times 2^{((n-69)/12)}$ , unde n este indexul MIDI al notei.
- **Selectare octavă:** Valoarea ADC (0-4095 pe ESP32) de la potențiometru este împărțită în intervale discrete corespunzătoare octavelor 2-7.
- **Generare ton PWM:** Funcția `ledcWriteTone(channel, frequency)` a ESP32 generează semnal PWM la frecvența calculată pe pinul buzzerului.
- **Debounce buton mute:** ISR-ul setează un flag, iar bucla principală verifică flag-ul cu `debounce software` (ex: 200ms între comutări).
- **Refresh OLED:** Display-ul este actualizat la fiecare ~100ms cu nota curentă, frecvența și octava, evitând flickering.

## Rezultate Obținute

Demo video al proiectului funcțional:

[Demo Video - Theremin Digital \(YouTube\)](#)

## Concluzii

În urma realizării proiectului, am aprofundat concepte de programare cu microprocesoare, în special lucrul cu PWM, ADC, I2C și întreruperi, aplicate într-un context practic. Principala dificultate a fost instabilitatea citirilor senzorului ultrasonic, rezolvată prin implementarea unui filtru de medie mobilă, și calitatea tonului generat de buzzer, îmbunătățită prin creșterea rezoluției PWM de la 8 la 12 biți.

A fost un proiect interesant de realizat și m-a ajutat să înțeleg mai bine cum se integrează mai multe periferice într-un sistem funcțional, nu doar individual ca în laborator. Pe viitor, proiectul ar putea fi extins cu un al doilea senzor pentru controlul volumului sau cu conectivitate Bluetooth pentru o aplicație mobilă.

## Download

Codul sursă complet al proiectului este disponibil pe GitHub:

[PM Fair - Theremin Digital \(GitHub Repository\)](#)

Arhiva cu surse, scheme și documentație va fi încărcată după finalizarea proiectului.

## Jurnal

- **Săptămâna 1 (28 apr - 4 mai):** Alegerea temei, documentare, comandă componente.
- **Săptămâna 2 (5 mai - 11 mai):** Completare documentație inițială pe OCW.

## Bibliografie/Resurse

### Resurse Hardware:

- [HC-SR04 Ultrasonic Sensor Datasheet](#)

- [SSD1306 OLED Display — Datasheet & Pinout](#)
- [Same Sky \(CUI Devices\) — CPT Series Piezo Transducers](#)
- [Espressif ESP32 — Specificații oficiale](#)

### Resurse Software:

- [Adafruit SSD1306 Library \(GitHub\)](#)
- [Adafruit GFX Library \(GitHub\)](#)
- [ESP32 Arduino Core — Documentație oficială](#)

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/bianca.popa1106/vlad.ilie1512>



Last update: **2026/05/25 12:32**