

# Tamagotchi Circuits

## Introducere

Tamagotchi Circuits este un proiect individual care combină embedded cu o interfață hardware interactivă pe ESP32.

Ideea principală: o fantomă (Ghost Boo) afișată pe un ecran OLED devine tristă în timp. Pentru a o menține fericită, utilizatorul trebuie să joace minigame-uri — unele dintre ele implică construirea unor circuite electronice reale pe breadboard, pe care ESP32 le validează și le face graficul pe un al doilea ecran OLED dedicat minigame-urilor.

Scopul proiectului este de a oferi un sistem de recompensă concret (menținem fantoma fericită) prin construirea și înțelegerea unor circuite electronice, învățând astfel concepte de bază din electronică și fizică.

Proiectul este finalizat cu 9 minigame-uri funcționale: potentiometer, RC circuit (6 configurații), voltage divider, diode rectifier, transistor switch, NPN amplifier, deadbeat rhythm, și metronome.

Abstractizarea generală a conceptului de „circuit minigame” ca struct ``minigame_t`` este implementată: fiecare minigame expune ``init``, ``deinit``, ``run`` și un ``mood_reward``, toate înregistrate într-un registry cu progresie liniară.

Proiectul este util pentru că stimulează dorința de a învăța electronică și de a construi circuite corecte, într-un mod interactiv și vizual.

## Descriere Generală

### Minigame-uri

#	Minigame	Descriere	GPIO-uri	Status
0	Potentiometer	Wiperul unui potențiomtru; menții zona țintă random 3s	GPIO32 (ADC1 CH4)	✓ Implementat
1	RC circuit (6 config.)	Construiești R+C pe breadboard; ESP măsoară $\tau$ prin least-squares pe $\ln(V)$	GPIO25 out, GPIO34 ADC	✓ Implementat
2	Voltage divider	Doi rezistori în serie; ESP validează raportul de tensiune	GPIO25 out, GPIO34 ADC	✓ Implementat
3	RC low/high-pass filter	R+C; ESP măsoară răspunsul în frecvență la PWM	GPIO25 PWM, GPIO34 ADC	TODO

4	Diode rectifier	Diodă + condensator pe semnal cosinus 200Hz redresat; osciloscop live	GPIO25 DAC, GPIO34 ADC	✓ Implementat
5	Capacitor buffer	Condensatorul menține tensiunea după tăierea GPIO-ului	GPIO25 out, GPIO34 ADC	TODO
6	Transistor switch	NPN în saturație (1kΩ bază); ESP conduce baza, măsoară Vc la colector	GPIO25 out (bază), GPIO34 ADC (colector)	✓ Implementat
6b	NPN Amplifier	NPN în regim activ (1MΩ bază); ESP măsoară β real (Ib=2.6μA fix)	GPIO25 out (bază), GPIO34 ADC (colector)	✓ Implementat
7	LM-35 temperature	Senzor temperatură extern, output direct pe ADC	GPIO35 (ADC1 CH7)	TODO
8	Deadbeet rhythm	ESP redă o secvență de LED-uri; utilizatorul o repetă pe butoane colorate	GPIO13/12/14 out (LED), GPIO19/17/16 in (R/G/B)	✓ Implementat
9	Metronome	Două potențiometre (BPM + duty cycle) + buzzer pasiv; ține 3s pentru pass	GPIO32, GPIO35 ADC; GPIO23 PWM	✓ Implementat

Games 1–6b partajează GPIO25 și GPIO34 — circuitul de pe breadboard se schimbă între ele, nu pinii.

## Schema bloc



Modulele principale și interacțiunile dintre ele:

- **ESP-32 Devkit V1** — nucleul sistemului; orchestrează toate minigame-urile, conduce ambele OLED-uri și perifericele
- **OLED SH1106 1.3"** — afișează fantoma permanent și starea ei (fericită / tristă / măsurând / celebrând)
- **OLED SSD1306 0.96"** — display dedicat minigame-urilor: instrucțiuni, bare de progres, osciloscop live
- **Buzzer pasiv (PWM)** — folosit în metronome minigame
- **LED-uri RGB** — folosite în deadbeet rhythm game
- **Butoane tactile** — input pentru deadbeet rhythm game + butonul principal BTN\_A (GPIO33)
- **ADC GPIO34** — măsurare circuite RC, voltage divider, rectifier, transistor, etc.
- **ADC GPIO35** — senzor temperatură LM-35 / duty cycle metronome
- **ADC GPIO32** — potențiometrul minigame / BPM metronome

## Hardware Design

### Pinout

## Display-uri

Display	Diagonală	Rol	Bus I2C	SDA	SCL	Adresă
SH1106	1.3"	Ghost — permanent activ	I2C0 (I2C_NUM_0)	GPIO18	GPIO5	0x3C
SSD1306	0.96"	Minigame support	I2C1 (I2C_NUM_1)	GPIO4	GPIO15	0x3C

Ambele display-uri au adresa 0x3C — menținute pe bus-uri separate pentru a evita conflictul. Ambele sunt cablate pe Side 1 al DevKit-ului pentru cablaj curat.

## Pini minigame-uri

#	Minigame	Circuit	Pini
0	Potentiometer	Wiper pot la ADC	GPIO32 (ADC1 CH4)
1	RC circuit	R + C la GND	GPIO25 out, GPIO34 ADC
2	Voltage divider	Doi rezistori în serie	GPIO25 out, GPIO34 ADC
3	RC low/high-pass filter	R+C, măsurare răspuns frecvență	GPIO25 PWM, GPIO34 ADC
4	Diode rectifier	Diodă + cap pe semnal cosine redresat	GPIO25 DAC, GPIO34 ADC
5	Capacitor buffer	Cap menține tensiunea după GPIO cut	GPIO25 out, GPIO34 ADC
6	Transistor switch	NPN saturație: 1kΩ bază, LED pe colector	GPIO25 out (bază 1kΩ), GPIO34 ADC (colector)
6b	NPN Amplifier	NPN activ: 1MΩ bază, 1kΩ colector, $\beta$ real	GPIO25 out (bază 1MΩ), GPIO34 ADC (colector)
7	Temperature LM-35	Output LM-35 la ADC	GPIO35 (ADC1 CH7)
8	Deadbeet rhythm	LED-uri secvență, butoane colorate repetă	GPIO13/12/14 out (R/G/B LED), GPIO19/17/16 in (R/G/B buton)
9	Metronome	Două pot-uri + buzzer pasiv	GPIO32 ADC (BPM), GPIO35 ADC (duty), GPIO23 PWM

## Pini ficși

Pin	Rol
GPIO33	BTN_A — buton principal (ISR, falling edge, debounce 300ms)
GPIO23	Buzzer PWM (metronome)

## Listă de piese

#	Componentă	Cantitate	Observații
1	ESP-32 Devkit V1 (DOIT)	1	Microcontroller principal
2	Ecran OLED SH1106 1.3", 128×64, I2C	1	SDA→GPIO18, SCL→GPIO5; display fantomă
3	Ecran OLED SSD1306 0.96", 128×64, I2C	1	SDA→GPIO4, SCL→GPIO15; display minigame
4	Senzor temperatură LM-35	1	ADC→GPIO35
5	Butoane tactile + set butoane colorate SYB-120	4+	3× deadbeet rhythm (R/G/B) + 1× BTN_A principal

6	LED-uri RGB	3	Deadbeet rhythm game (R/G/B)
7	Buzzer pasiv	1	Metronome minigame (PWM GPIO23)
8	Potențiometre	2+	1× potentiometer minigame (GPIO32), 1× metronome duty (GPIO35); potențiometre colorate SYB-120
9	Rezistențe, condensatoare, diode	asortate	Pentru circuit minigame-uri (RC, divider, filter, rectifier)
10	Tranzistoare NPN 2N2222	2+	Transistor switch minigame (1kΩ bază) + NPN amplifier (1MΩ bază)
11	Kit breadboard SYB-120 (Optimus Digital)	1	4 breadboard-uri, fire, butoane colorate, potențiometre
12	Fire de breadboard	—	Conectică generală

## Scheme electrice

[Schema hardware placa \(PDF\)](#)

**Voltage divider, RC circuit diode rectifier, transistor switch, NPN amplifier, deadbeet rhythm, metronome:** Vezi caietul cu ecuațiile și schemele pentru referință.

**TODO:** Schemă electrică pentru RC low/high-pass filter

**TODO:** Schemă electrică pentru capacitor buffer

**TODO:** Schemă electrică pentru LM-35 temperature sensor

## Diagrame de semnal

Vezi caietul cu ecuațiile și schemele pentru referință.

## Software Design

### Mediu de dezvoltare

- PlatformIO + ESP-IDF v5.x (framework-espidf)
- Toolchain: xtensa-esp-elf
- Librării third-party: niciuna — drivere custom pentru SH1106 și SSD1306, ADC oneshot (ESP-IDF built-in)

### Arhitectură

## Două task-uri FreeRTOS:

- `ghost\_task` — 8KB stack, rulează permanent la 50ms tick; gestionează animațiile fantomei, decayul de mood și spawn-ul minigame\_task
- `minigame\_task` — 4KB stack, spawnat on-demand la apăsarea butonului; distrus la finalul minigame-ului

## Abstractizare minigame (`minigame\_t`):

```
typedef struct {
    const char *name;           // afișat pe SH1106 „NEXT:”
    esp_err_t (*init)(void);    // configurare GPIO/ADC
    void (*deinit)(void);      // restaurare GPIO safe state
    bool (*run)(void);         // loop blocant; true = pass
    uint8_t mood_reward;       // puncte mood la PASS
} minigame_t;
```

**Registry (`minigame\_registry.c`):** progresie liniară; `registry\_advance()` apelat indiferent de rezultat (pass sau abort).

**Ghost state machine:** 5 stări — NEUTRAL, SAD, HAPPY, CELEBRATING, MEASURING. Decay de mood dependent de bandă (mai rapid la mood scăzut). Animații 2-frame per stare pe SH1106.

**Buton (`g\_minigame\_btn`):** ISR pe GPIO33 (falling edge, debounce 300ms). Când un minigame este activ, apăsările sunt rutate prin `g\_minigame\_btn` — fiecare minigame își gestionează propria logică de stop/avans.

## RC Minigame — detalii algoritm

6 configurații modulare, toți parametrii derivați din `tau\_ms`:

Config	R	C	$\tau$	Concept
RC 100ms	1k $\Omega$	100 $\mu$ F	100ms	Rezistență mică → descărcare rapidă
RC 200ms	2k $\Omega$	100 $\mu$ F	200ms	R dublu → $\tau$ dublu
RC 500ms	10k $\Omega$	2×100 $\mu$ F serie	500ms	C serie reduce capacitatea
RC 1s	10k $\Omega$	100 $\mu$ F	1s	Configurație de bază
RC 2s	10k $\Omega$	2×100 $\mu$ F paralel	2s	C paralel dublează capacitatea
RC 4s	20k $\Omega$ (2× serie)	2×100 $\mu$ F paralel	4s	Compus — $\tau = R \times C$ scalează cu ambii

- `charge\_ms = 6 ×  $\tau` (încărcare la 99.75% din Vs)$
- `sample\_interval\_ms = 5 $\tau$  / 128` (128 sample-uri acoperă 5 $\tau$ )
- $\tau \geq 500$ ms → osciloscop live (flush per sample);  $\tau < 500$ ms → batch (toate sample-urile, un singur flush)
- $\tau$  calculat prin least-squares pe  $\ln(V(t))$  cu timestamps reale (`esp\_timer\_get\_time`)
- Toleranță  $\pm 35\%$  pentru a acomoda toleranța componentelor electronice (condensatoare  $\pm 20\%$ )

## TODO Software

- RC low/high-pass filter minigame
- Capacitor buffer minigame
- LM-35 temperature minigame

## Concluzii

Proiectul a atins obiectivul principal: 9 minigame-uri funcționale, fiecare cerând utilizatorului să construiască un circuit real pe breadboard și să îl valideze cu ESP32.

Abstractizarea `minigame\_t` s-a dovedit robustă — fiecare joc nou a necesitat modificări în exact 3 fișiere (minigame nou, `minigame\_registry.c`, `minigame\_rewards.h`), fără a atinge codul existent.

Cea mai complexă măsurătoare: RC  $\tau$  prin least-squares pe  $\ln(V)$  cu timestamps reale, toleranță  $\pm 35\%$  pentru componente reale. Cea mai vizuală: diode rectifier cu osciloscop live pe SSD1306 — 128 sample-uri la 200Hz, 2 cicluri complete pe ecran.

GPIO25 și GPIO34 sunt partajate de 7 jocuri (RC, voltage divider, rectifier, transistor switch, NPN amplifier) — breadboard-ul se schimbă între minigame-uri, pinii nu. Aceasta a simplificat semnificativ cablajul permanent.

Remaparea display-urilor pe Side 1 al DevKit-ului (SSD1306→GPIO4/15, SH1106→GPIO18/5) a eliminat firele lungi care traversau breadboard-ul.

Rămân de implementat: RC filter, capacitor buffer și LM-35 — fiecare necesită aproximativ aceeași structură ca jocurile existente.

## Download

**TODO:** Arhivă cu sursele finale, README și instrucțiuni de compilare/flash.

## Jurnal

- **11 aprilie** — prima comandă: breadboard, SSD1306 (OLED 0.96" I2C), LM-35
- **6 mai** — a doua comandă: ESP-32, SH1106 (OLED 1.3" I2C)
- **10 mai** — primul demo funcțional: fantoma animată pe OLED, circuit RC pe breadboard, grafic al curbei de descărcare afișat pe ecran
- **16 mai** — sistem complet de abstractizare minigame (minigame\_t, registry, două task-uri FreeRTOS); potentiometer minigame funcțional; RC circuit minigame: osciloscop live pe SSD1306,  $\tau$  prin least-squares, 6 configurații modulare
- **18 mai** — deadbeet rhythm minigame implementat: secvență LED R/G/B generată de ESP,

utilizatorul repetă pe 3 butoane

- **20 mai** — achiziție kit SYB-120 (4 breadboard-uri, butoane colorate, potențiometre) de la Optimus Digital
- **23 mai** — voltage divider minigame implementat
- **23 mai** — remapare display-uri pe Side 1: SSD1306→GPIO4/GPIO15, SH1106→GPIO18/GPIO5; butoane deadbeet remapate pe GPIO19/17/16
- **23 mai** — metronome minigame implementat: două potențiometre (BPM + duty cycle) + buzzer pasiv, hold 3s pentru pass
- **23 mai** — diode rectifier minigame implementat: DAC cosine 200Hz pe GPIO25, osciloscop live SSD1306, validare prin std dev < 100mV
- **24 mai** — NPN transistor switch minigame implementat (1kΩ bază, LED pe colector, pass dacă  $V_c/V_{cc} < 0.15$ ); adăugat NPN\_THEORY.md cu pinout 2N2222
- **26 mai** — NPN amplifier variant implementat: 1MΩ bază ( $I_b=2.6\mu A$ ), SSD1306 afișează  $V_c$ ,  $I_c$  și  $\beta$  real; pass dacă  $20 \leq \beta \leq 800$

## Bibliografie/Resurse

- Datasheet ESP32 — [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)
- Datasheet SH1106 — <https://www.displayfuture.com/Display/datasheet/controller/SH1106.pdf>
- Datasheet SSD1306 — <https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf>
- Datasheet LM-35 — <https://www.ti.com/lit/ds/symlink/lm35.pdf>
- Datasheet 2N2222 — <https://www.onsemi.com/pdf/datasheet/p2n2222a-d.pdf>
- ESP-IDF v5 I2C Master docs — <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/i2c.html>
- ESP-IDF v5 ADC Oneshot docs — [https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/adc\\_oneshot.html](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/adc_oneshot.html)
- ESP-IDF v5 DAC Cosine docs — <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/dac.html>
- ESP-IDF v5 FreeRTOS docs — <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/freertos.html>

[Export to PDF](#)

From:

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

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2026/theodor\\_ioan.buliga/ioan.sabou](http://ocw.cs.pub.ro/courses/pm/prj2026/theodor_ioan.buliga/ioan.sabou)



Last update: **2026/05/26 23:12**