



Youtube

Engineer-Amel

Youtube channel @engineeramel

ESP32 BLOCK BREAKER GAME
WITH BLUETOOTH CONTROL

<https://www.youtube.com/@engineeramel>

YOUTUBE

2025






BBreakerX – ESP32 Breakout Game

An Arduino-based retro-style Breakout game built for ESP32 with OLED display, Bluetooth controls via the Dabble app, and built-in sound & effects.

Gameplay Features

Feature	Description	
Classic Breakout	Bounce a ball to destroy bricks using a paddle.	
3 Difficulty Levels	Easy, Normal, Hard — affects paddle size and ball speed.	
Paddle Control	Move left/right using Dabble GamePad over Bluetooth.	
Score System	Gain 10 points per brick; score displayed in-game.	
Lives System	Start with 5 lives; lose one if all balls fall off-screen.	
Victory/Defeat States	Win by clearing all bricks; lose if lives reach zero.	

Game Mechanics

Mechanic	Description
Multiball Support	Up to 2 active balls; triggered by power-up.
Power-Ups	3 Types: <ul style="list-style-type: none">♦ Paddle extender♦ Extra life♦ Multiball
Brick Grid	4 Rows × 6 Columns — resets each game.
Particle Effects	Simple visual effects on brick break using falling pixels.



BBreakerX – ESP32 Breakout Game

An Arduino-based retro-style Breakout game built for ESP32 with OLED display, Bluetooth controls via the Dabble app, and built-in sound & effects.



OLED Display Features (SSD1306)

Visual Element	Description
Main Menu	Displays game title and difficulty selection.
Bricks & Paddle	Drawn as rectangles and rounded bars.
Balls	Circular pixels representing balls.
Power-Ups	Small 6×6 pixel square icons.
Score/Lives HUD	Displayed at the top-left corner of screen.



Sound Features

Event	Sound Effect
Wall Bounce	Medium pitch beep
Paddle Hit	High pitch tone
Brick Break	Sharp tone + particles
Life Lost	Low tone indicating failure
Win Sound	Victory tone when all bricks are cleared
Background Loop	Alternating tones every 100 ms while playing



BBreakerX – ESP32 Breakout Game

An Arduino-based retro-style Breakout game built for ESP32 with OLED display, Bluetooth controls via the Dabble app, and built-in sound & effects.



Dabble App Controls

Connect via Bluetooth using [Dabble](#) app:

Button	Function
Up/Down	Navigate difficulty options in menu
Left/Right	Move paddle during gameplay
Select	Confirm selection / start game



Technical Features


Feature	Description
Non-blocking Sound	Background tones using <code>millis()</code> timing.
Object Pooling	Pre-allocated arrays for balls, power-ups, and particles.
Modular Codebase	Clean separation of setup, logic, and rendering functions.

hardware and software components:

Control via Dabble App


Item	Description
Dabble App	Smartphone app (by STEMpedia) to emulate a gamepad via Bluetooth
Bluetooth Classic	Communication method between ESP32 and phone via <code>DabbleESP32.h</code>
GamePad Module	Virtual D-Pad + Select button in Dabble used for controlling paddle and menus

Software Libraries

Library	Use	
<code>Adafruit_SSD1306</code>	Drives the 128x64 OLED screen via SPI or I2C	
<code>Adafruit_GFX</code>	Basic graphics library for drawing text/shapes	
<code>SPI.h</code>	SPI protocol to communicate with OLED	
<code>DabbleESP32.h</code>	Enables ESP32 to connect and receive input from Dabble GamePad	
<code>ledcWriteTone()</code> (ESP32)	Used for sound output with non-blocking tone generation	

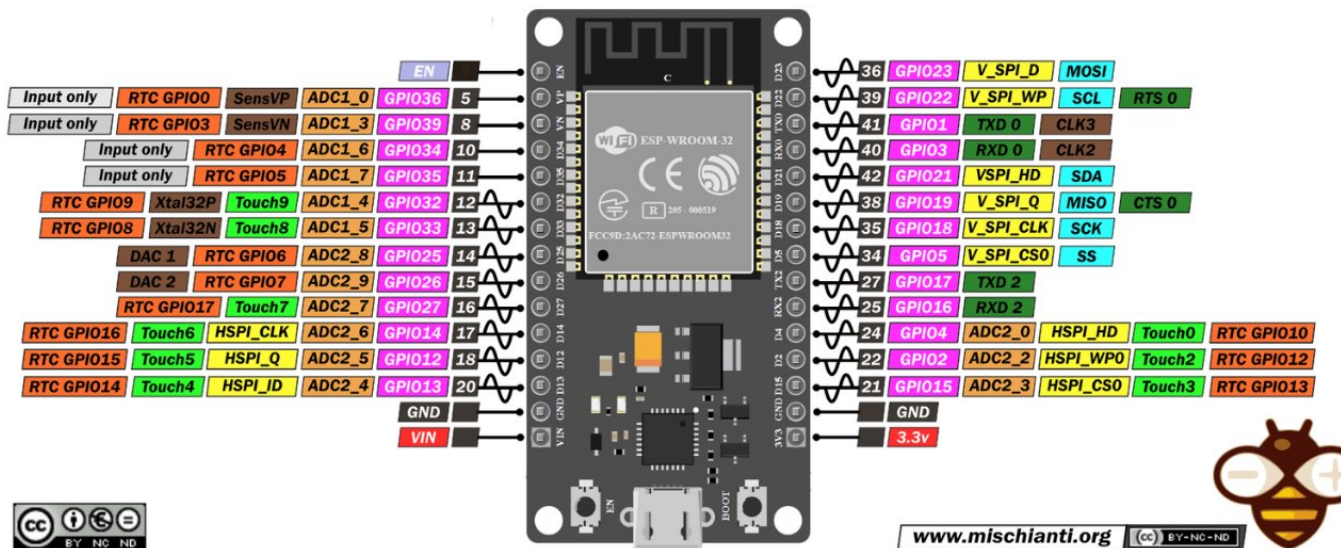
Summary of Required Parts

Here's a parts list you can use:

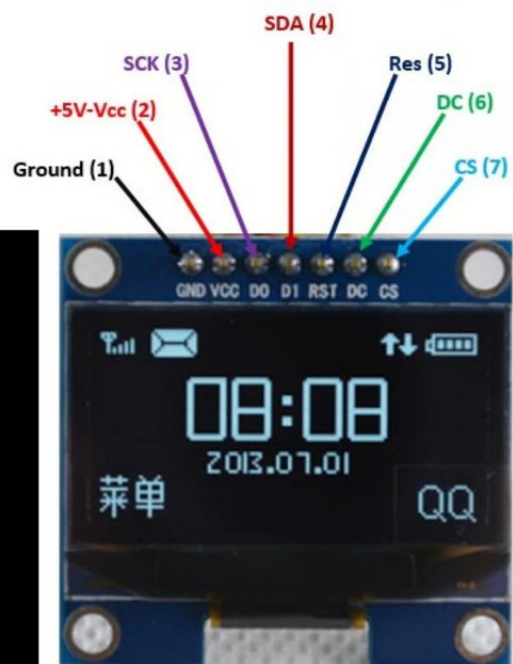
Part	Quantity	
ESP32 Dev Board (e.g., NodeMCU)	1	
SSD1306 OLED Display (128x64 SPI)	1	
Passive Buzzer or Speaker	1	
Resistor (100–220Ω for speaker)	1	
Jumper Wires	Several	
Breadboard (optional)	1	
Power Supply (USB/Battery)	1	
Smartphone (for Dabble app)	1	

main compounds

ESP32 DEV KIT V1 PINOUT



Here's the OLED type used:



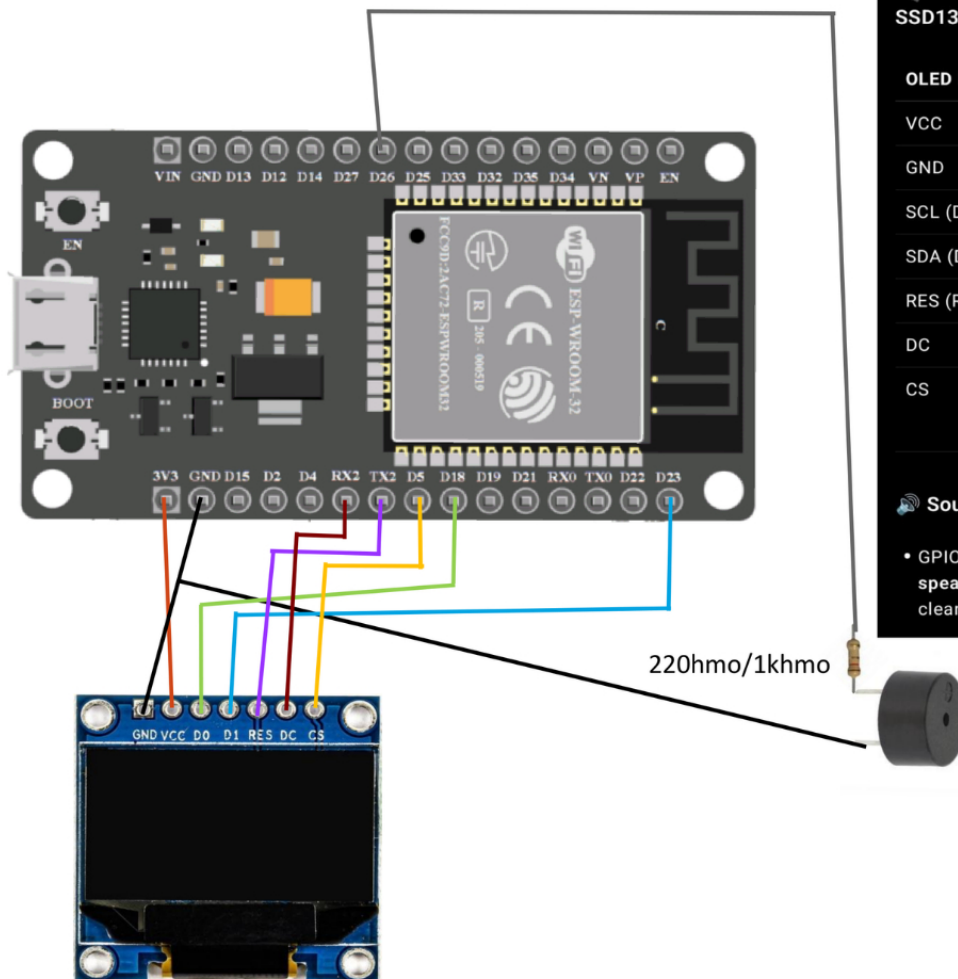
Display Type: SSD1306

Resolution: 128 × 64 pixels

Interface: SPI (Serial Peripheral Interface)

Library Used: Adafruit_SSD1306

Circuit diagram



Hardware Wiring (ESP32 to OLED SSD1306 – 7-Pin SPI):

OLED Pin	ESP32 Pin	Function
VCC	3.3V	Power
GND	GND	Ground
SCL (D0)	18	SPI Clock
SDA (D1)	23	SPI MOSI
RES (RST)	17	Reset Pin
DC	16	Data/Command
CS	5	Chip Select

Sound DAC Pin:

- GPIO 25 → Connect to **Amplifier or small speaker** via resistor-capacitor filter for clean DAC output.

Using App

Playstore

Step : 1
download app in playstore



Arduino ESP Bluetooth
- Dabble
STEMpedia

4.3 ★
1K reviews ⓘ

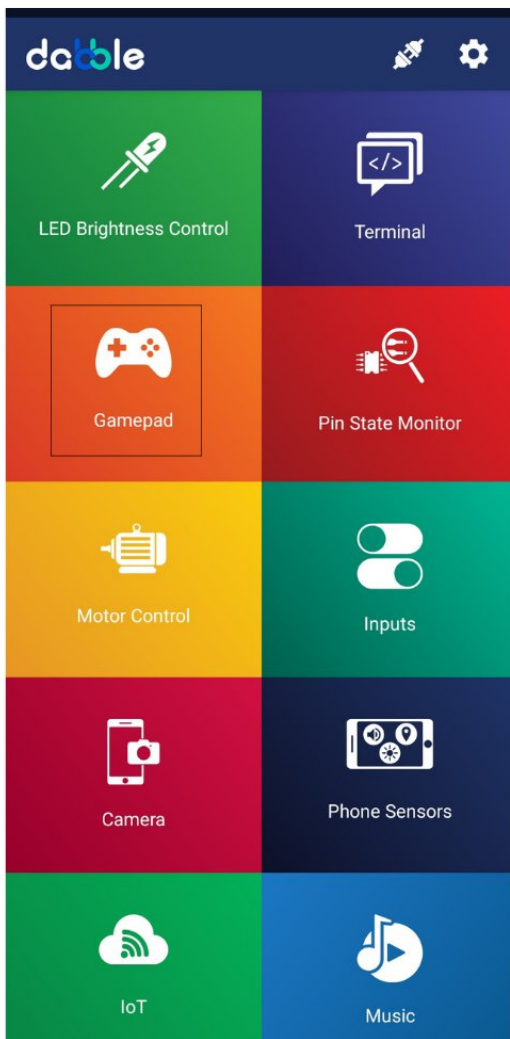
3+
Rated for 3+ ⓘ

1L+
Downloads

Uninstall

Open

Step : 2
select Gamepad



Step : 3
to select esp32 drive (BBreakerX) and
pairing to Bluetooth



Important!

<https://github.com/STEMpedia/DabbleESP32>



```
1 #include <SPI.h>
2 #include <Adafruit_GFX.h>
3 #include <Adafruit_SSD1306.h>
4 #include <DabbleESP32.h>
5
6 // Screen and Display settings
7 #define SCREEN_WIDTH 128
8 #define SCREEN_HEIGHT 64
9 Adafruit_SSD1306 display(SCREEN_WIDTH,
10 SCREEN_HEIGHT, &SPI, 16, 17, 5); //
   Adjust pins for your setup
```

DabbleESP32 1.5.1

this library install only the sketch program code work properly

DabbleEsp32.h library download for this link
<https://github.com/STEMpedia/DabbleESP32>