**Pinout (ESP32 WROOM / NodeMCU ESP32S)**

* **I²C (3.3 V lógica)**
  + SDA → GPIO 21
  + SCL → GPIO 22
  + Dispositivos en el bus: SHT31 (0x44) y PCA9685 (0x40)

* **Sensores digitales (entradas)**
  + IR obstáculo (hay objeto) → GPIO 26 (entrada digital)
  + Inductivo NPN LJ12A3-4-Z/BX (detectar metal) → GPIO 27 (entrada digital, ver adaptación a 3.3 V abajo)
  + Endstop X\_min (NC, activo-bajo) → GPIO 16 (entrada con INPUT\_PULLUP)

* **Señalización / actuadores lógicos**
  + NeoPixel DIN (vía 74HCT14 + R serie 330–470 Ω) → GPIO 23 (salida)  
     Buzzer (a través de NPN + R base) → GPIO 2 (salida)

* **Balanzas / llenado (3 tachos)**
  + HX711 SCK (común a los 3 módulos) → GPIO 25 (salida)
  + HX711 DOUT Tacho 1 (Metal) → GPIO 34 (solo entrada)
  + HX711 DOUT Tacho 2 (Orgánico) → GPIO 35 (solo entrada)
  + HX711 DOUT Tacho 3 (Resto) → GPIO 36 (solo entrada)

* **Eje lineal (ULN2003 + paso a paso 6 hilos)**
  + ULN2003 IN1 → GPIO 18 (salida)
  + ULN2003 IN2 → GPIO 19 (salida)
  + ULN2003 IN3 → GPIO 5 (salida)
  + ULN2003 IN4 → GPIO 17 (salida)

*(En firmware se usa AccelStepper FULL4WIRE con orden IN1,IN3,IN2,IN4.)*

* **Servo “gate” (abre compuerta)**
  + PCA9685 canal CH0 → microservo de compuerta (PWM 50 Hz)

Alimentación y dominios

* 3.3 V (lógica, desde el ESP32): ESP32, SHT31, PCA9685 (VCC, no V+), HX711 (recomendado a 3.3 V para compatibilidad de niveles).
* 5 V (potencia liviana): NeoPixel, V+ de PCA9685 (servo gate).
* 9–12 V (motor paso a paso): alimentación del motor al ULN2003 (recomendado 9 V según tu motor).
* 12 V (sensor inductivo NPN): el LJ12A3 suele requerir 6–36 V. 12 V es típico.

**Masa común entre todas las fuentes (3.3 V, 5 V, 9/12 V) y retorno corto.**

**Desacople: 100 nF por IC; 1000 µF entre 5 V y GND cerca de la tira NeoPixel; 100–220 µF cerca del ULN2003.**

**Protección: fusible/PTC en 5 V (y en 9/12 V si es posible).**

Paso a paso de conexión

**1) Preparar fuentes y masas**

1. Colocá tus fuentes: 3.3 V (ESP32), 5 V ≥ 3 A (NeoPixel/servo), 9 V (motor), 12 V (sensor inductivo).
2. Uní todas las GND en un punto común (estrella) para evitar lazos.
3. Agregar 1000 µF/10 V entre 5 V y GND junto a la tira NeoPixel; 100 nF cerca de cada PCA9685, SHT31 y cada HX711.
4. Verificar polaridades y continuidad.

**2) Bus I²C (SHT31 + PCA9685)**

1. Conectar SDA (GPIO21) y SCL (GPIO22) del ESP32 a SDA/SCL de SHT31 y PCA9685 en paralelo.
2. VCC (3.3 V) a VCC de ambos; GND común.
3. En el PCA9685:
   1. VCC = 3.3 V (lógica I²C).
   2. V+ (servo power) = 5 V (no lo mezcles con VCC).
   3. GND de V+ y GND de lógica deben estar en común.
4. Direcciones por defecto: SHT31 0x44, PCA9685 0x40

**3) Servo “gate” (PCA9685 CH0)**

1. Conectar el microservo (señal al pin CH0 del PCA9685, rojo a 5 V+, negro a GND).
2. En el firmware ya están mapeados los pulsos (≈500–2400 µs mapeados a ticks del PCA).

**4) NeoPixel con 74HCT14 (3.3→5 V)**

* ESP32 GPIO23 → R serie 47 Ω → DIN de la tira.
* Tira NeoPixel: 5 V y GND robustos; 1000 µF cerca; no alimentarla desde el ESP32.
* GND de la tira común con el ESP32.

**5) Buzzer 5 V con NPN**

1. ESP32 GPIO14 → R base 1 kΩ → base de 2N2222/SS8050.
2. Emisor a GND; colector al terminal negativo del buzzer; terminal positivo a 5 V.
3. Si el buzzer es inductivo (no piezo activo), agregar diodo 1N4148/1N5819 en antiparalelo (cátodo a +5 V).

**6) Sensores de presencia/clasificación**

**6.1 IR de obstáculo (trigger de ventana de muestreo)**

1. VCC: preferí 3.3 V (si el módulo lo soporta) para que la salida sea 3.3 V segura.  
    Si usás 5 V, bajá la salida a 3.3 V con un divisor 10 k/20 k o transistor.
2. GND común.
3. OUT → GPIO26.

**6.2 Inductivo NPN LJ12A3-4-Z/BX (detectar metal)**

1. Marrón a +12 V, Azul a GND.
2. Negro (OUT): es colector abierto. Dos opciones seguras:
   1. Adaptación simple: OUT al GPIO27 con pull-up de 10 kΩ a 3.3 V (y una R serie 1 kΩ al GPIO). *GND en común con 12 V y 3.3 V.*
   2. Aislamiento (recomendado): OUT → optoacoplador → ESP32 (3.3 V).
3. Lógica en firmware: HIGH = metal (active-high tras el pull-up a 3.3 V).

**6.3 Endstop X\_min (NC, activo-bajo)**

1. Común del switch a GND; contacto NC al GPIO16.
2. Configurar el pin con INPUT\_PULLUP (ya está en firmware).
3. Cuando “toca”, el pin lee LOW.

**7) Balanzas (3× HX711 + celdas de carga)**

1. Usar 1 HX711 por tacho. Aquí se asume 3 módulos.
2. Alimentación HX711: 3.3 V + GND (para que DOUT/SCK estén en 3.3 V).
3. Señales:
   1. SCK (común) → GPIO25.
   2. DOUT Tacho 1 (Metal) → GPIO34.
   3. DOUT Tacho 2 (Orgánico) → GPIO35.
   4. DOUT Tacho 3 (Resto) → GPIO36.
4. Celdas: cable corto, fijación rígida, sin cargas laterales. A+ / A- a la galga; E+ / E- al puente.
5. Calibración: offset con tacho vacío; escala con masa patrón; guardar en NVS o en firmware.

**8) Motor paso a paso 6 hilos + ULN2003 (eje horizontal)**

1. Motor 6 hilos unipolar (dos comunes al +V, 4 bobinas a colectores). En la placa ULN2003:
   1. Conecta las 4 bobinas del motor a las salidas del ULN2003 (OUT1..OUT4).
   2. Uní los dos comunes del motor al +9 V (o el valor nominal de tu motor).
2. Entradas ULN2003 desde el ESP32:
   1. IN1 ← GPIO18
   2. IN2 ← GPIO19
   3. IN3 ← GPIO5
   4. IN4 ← GPIO17
3. GND del ULN2003 común con el ESP32 y con la fuente del motor.
4. Alimentación del motor: +9 V al pin VM del ULN2003 / común del motor.
5. Condensador de 100–220 µF cerca de la bornera del ULN2003.
6. En firmware se usa AccelStepper (FULL4WIRE, IN1, IN3, IN2, IN4) para el orden correcto y movimiento no bloqueante.

**9) Distribución física / posiciones**

1. Distancia entre centros de tachos: 12 cm (120 mm).
2. Anchura de cada tacho: ~10 cm.
3. pos\_mm\_bins (firmware / API): {0, 120, 240} mm, medidos desde el cero hecho por homing (X\_min).
4. En homing: el eje baja a X\_min, libera (bounce), “kiss” lento, fija posición 0 y se posiciona al centro del Tacho 1.

Chequeo final

* **Fuentes separadas: 3.3 V (lógica), 5 V (NeoPixel/servo), 9 V motor, 12 V inductivo. GND común.**
* **NeoPixel con R 47 Ω en DIN - OUT→DIN del otro** **Neopixel.**
* **Inductivo con pull-up a 3.3 V + R serie 1 kΩ u opto (recomendado).**
* **HX711 a 3.3 V; SCK común (GPIO25); DOUT en GPIO34/35/36.**
* **I²C correcto: SDA21/SCL22 → SHT31 (0x44) y PCA9685 (0x40).**
* **ULN2003 IN1..IN4 en 18,19,5,17; motor a +9 V; GND común.**
* **Endstop X\_min NC a GPIO16 con INPUT\_PULLUP.**
* **Servo gate en PCA9685 CH0; V+ 5 V para potencia de servo.**
* **Verificar pos\_mm\_bins = [0,120,240] en dashboard/API/firmware.**