The Ultimate Animatronic Endoskeleton Project Plan

This plan will guide you through building a wirelessly controlled animatronic endoskeleton with blinking eyes, 3-axis eye movement, a 2-axis torso, and waving hands.

Phase 1: Design & Mechanical Planning (The Endoskeleton)

This phase focuses on the visible structure and how each moving part will be physically constructed and actuated.

1. Conceptual Endoskeleton Design

- **Objective:** Visualize the animatronic's form, scale, and how the internal mechanisms (servos, linkages) will be part of the aesthetic.
- **Actionable Steps:**
    - **Sketch the Endoskeleton:** Draw it from multiple angles. Since it's an endoskeleton, think about exposed "bones" and joints. Will it have a head, neck, torso, and arms? How articulated will each section be?
    - **Determine Scale:** A tabletop size (e.g., 1-2 feet tall) is manageable with your current servos. Larger might require stronger servos.
    - **Joint & Linkage Design:** For each movement (eye pan/tilt/roll, blink, torso pan/tilt, hand wave), sketch how the servo will physically connect to the "bone" or moving part. Consider using skewers, paperclips, and hot glue to form levers, pushrods, and pivots.
        - **Eye Mechanism:** A "gimbal" style using skewers as pivot points for pan, tilt, and roll. The servos will push/pull these skewers.
        - **Blinking:** A simple lever mechanism connected to a servo, moving a lightweight eyelid (e.g., small piece of paper or thin card).
        - **Torso:** A base pivot for horizontal rotation (pan) and a higher pivot/hinge for forward/backward tilt.
        - **Hands:** A pivot at the "wrist" for the waving motion.
    - **Component Housing:** Plan where the **Animatronic Brain ESP32**, **PCA9685**, and servo power supply will be integrated into the endoskeleton's structure. They should be accessible but out of the way of movements.
- **Deliverables:** Detailed sketches showing the endoskeleton's overall structure, specific joint designs, and component placement.

2. Eye Mechanism Prototype

- **Objective:** Build a functional, visible prototype of the blinking and 3-axis eye movement. This is typically the most complex part.
- **Actionable Steps:**
    - **"Eyeball" Fabrication:** Use a lightweight sphere (e.g., ping pong ball, craft foam ball, or a well-rolled paper ball) as the eyeball.
    - **3-Axis Structure:**
        - Create a simple, sturdy frame using skewers and hot glue that cradles the eyeball.

        - **Pan:** Mount one servo to rotate the entire eye assembly horizontally.
        - **Tilt:** Mount a second servo to tilt the eye assembly vertically within the pan mechanism.
        - **Roll:** Mount a third servo to rotate the eyeball on its own axis. This might involve a small ring or cradle around the eyeball, connected to the servo.
    - **Blinking Mechanism:** Mount a fourth servo. Attach a small, lightweight "eyelid" (paper, thin plastic) to a lever connected to this servo, positioned to slide over/uncover the eyeball.
    - **Servo Mounting:** Securely attach the 4 eye servos to this prototype frame using paperclips, zip ties, and hot glue. Ensure their movements are smooth and unobstructed.
    - **Initial Servo Test:**
        - Connect the **PCA9685** to your **Animatronic Brain ESP32** (SDA/SCL, VCC, GND).
        - Connect your dedicated **5V servo power supply** to the PCA9685's V+ and GND terminals, ensuring a **common ground** with the ESP32.
        - Connect the 4 eye servos to PCA9685 channels (e.g., 0-3).
        - You'll use MicroPython for this. Load a basic script to the ESP32 to individually test each servo's movement, identifying their precise minimum and maximum PWM values for smooth, non-straining operation.
- **Deliverables:** A working prototype of the eye mechanism with 4 servos, initial MicroPython code for testing and calibration of these servos.

3. Torso & Hand Mechanism Prototypes

- **Objective:** Build functional, visible prototypes for the torso (pan/tilt) and hand (waving) movements.
- **Actionable Steps:**
    - **Torso Pan:** Create a sturdy base. Mount one servo to rotate the entire torso structure horizontally. A large lazy Susan bearing or a simple pivot with good support can work.
    - **Torso Tilt:** Create a hinge joint (using skewers, paperclips, or thin cardboard) higher up on the torso. Mount a second servo to actuate this hinge, tilting the upper torso forward and backward.
    - **Arm & Hand Structure:** Construct simple "bone" segments for the upper arm and forearm using skewers or layered cardboard.
    - **Hand Waving:** Create a simple pivot at the "wrist" joint. Mount one servo per hand at this pivot point to drive the waving motion.
    - **Initial Servo Test:** Connect these 4 servos (2 torso, 2 hand) to available PCA9685 channels (e.g., 4-7). Add them to your existing ESP32 MicroPython script and calibrate their min/max PWM values for their respective movements.
- **Deliverables:** Working prototypes of the torso and hand mechanisms with 4 servos, updated MicroPython code with calibrated values.

Phase 2: Endoskeleton Construction & Electronics Integration

This phase brings all your prototypes together into the final animatronic structure and handles the full wiring.

1. Main Endoskeleton Frame Assembly

- **Objective:** Build the complete, sturdy endoskeleton structure that showcases the mechanical articulation.
- **Actionable Steps:**
    - **Material Selection:** Use sturdy materials for the "bones." Options include:
        - **Thick Cardboard/Foam Board:** Layer multiple pieces for strength.
        - **Wooden Dowel Rods/ Bamboo Skewers (various diameters):** Excellent for structural members and pivots.
        - **Thin PVC Pipe:** Lightweight and strong, good for limbs.
        - **Plastic Sheets (e.g., from old containers):** Can be cut and shaped for plates and brackets.
    - **Stable Base:** Construct a wide, weighted base. This is critical to prevent tipping, especially as the animatronic moves.
    - **Torso Integration:** Securely attach the torso pan and tilt mechanisms to the base. Ensure smooth, unobstructed movement.
    - **Arm Integration:** Attach the upper arms to the torso. Even if you're not articulating the elbows/shoulders yet, ensure the arms move freely with the torso and don't impede its movement.
    - **Hand Integration:** Attach the forearm/hand assemblies, incorporating the waving servos, to the upper arms.
    - **Head/Neck Integration:** Securely attach the eye mechanism assembly to the top of the torso/neck. Design this so the eye mechanism can either move with the torso, or be independently controlled if you decided to add a neck servo (though with 8 servos total, prioritize eyes, torso, and hands).
    - **Servo Mounting:** Permanently mount all 10 servos to their designated locations within the endoskeleton frame. Use hot glue reinforced with paperclips or small zip ties for strong, permanent bonds.
- **Deliverables:** A complete, articulated endoskeleton frame with all servos securely mounted in place.

2. Electronics Mounting & Wiring

- **Objective:** Neatly and securely mount all electronic components and perform the final wiring, ensuring reliability and proper power delivery.
- **Actionable Steps:**
    - **PCA9685 & Animatronic Brain ESP32 Mounting:** Choose a discrete but accessible location within or on the endoskeleton (e.g., lower back of torso, or inside the base). Securely mount the **PCA9685** and the **Animatronic Brain ESP32** using hot glue, small zip ties, or custom-made holders.
    - **Servo Wiring (to PCA9685):**
        - Connect all 10 **servos** to the PCA9685's channels.
        - For each servo, connect the **Signal (Orange/Yellow/White) wire** t o the **'S' pin**, the **Red (VCC) wire** to the **'+' pin**, and the **Brown/Black (GND) wire** to the **'-' pin** of its assigned channel.

        - Refer to your calibrated servo-to-channel mapping from Phase 1.
    - **PCA9685 Power & Ground:**
        - Connect the **VCC pin of the PCA9685** to the **3.3V pin of the Animatronic Brain ESP32**. (Powers the PCA9685 chip).
        - Connect the **GND pin of the PCA9685** to a **GND pin of the Animatronic Brain ESP32**.
        - Connect your **dedicated 5V servo power supply's positive (+)** to the **V+ terminal block on the PCA9685**.
        - Connect your **dedicated 5V servo power supply's negative (-)** to the **GND terminal block on the PCA9685**.
    - **Common Ground (CRITICAL!):** Run a jumper wire from any **GND pin on the PCA9685 terminal block** (which is connected to your servo power supply's GND) to a **GND pin on your Animatronic Brain ESP32**. This creates a common ground for the entire animatronic's electronics.
    - **I2C Communication (Animatronic Brain ESP32 to PCA9685):**
        - Connect **SDA (PCA9685)** to **GPIO 21 (SDA) on the Animatronic Brain ESP32**.
        - Connect **SCL (PCA9685)** to **GPIO 22 (SCL) on the Animatronic Brain ESP32**.
    - **Controller ESP32 & Joystick Wiring:**
        - Mount your **Joystick** and the **Controller ESP32** onto a small breadboard or within a simple cardboard controller housing.
        - **Joystick VCC:** Connect to the **3.3V pin on the Controller ESP32**.
        - **Joystick GND:** Connect to a **GND pin on the Controller ESP32**.
        - **Joystick VRx (X-axis):** Connect to **GPIO 34 (Analog Input) on the Controller ESP32**.
        - **Joystick VRy (Y-axis):** Connect to **GPIO 35 (Analog Input) on the Controller ESP32**.
        - **Joystick SW (Button):** Connect to **GPIO 27 (Digital Input) on the Controller ESP32**.
        - **Joystick 2 VCC And GND common ESP32 3V3 and GND** with **Joystick 1**
        - **Joystick 2 VRx (X-axis):** Connect to **GPIO 36 (Analog Input) on the Controller ESP32**.
        - **Joystick 2 VRy (Y-axis):** Connect to **GPIO 39 or any other analog input pin (Analog Input) on the Controller ESP32**.
        - **Joystick 2 SW (Button):** Connect to **any Digital Input pin on the Controller ESP32**.
    - **Power or Controller ESP32:** Power it via its USB port for simplicity and portability.
- **Deliverables:** A fully wired endoskeleton with securely mounted ESP32 and PCA9685, and a separate, fully wired joystick controller.

Phase 3: Software Development (MicroPython Code)

This phase involves programming both ESP32s using MicroPython. You will need to flash MicroPython firmware onto both ESP32s first. Tools like `esptool.py` and `Thonny IDE` are excellent for this.Code is already given in the repo.