Robotic Arm Wiring Architecture Detailed Explanation in English

Here is the complete Wiring Architecture of your system:

1. Power Supply Distribution (Most Important Rule)

In Robotics, 90% of failures happen because of incorrect power wiring.

- **The Connection:**
    - ESP32 is powered through a USB cable or 5V Adapter.
    - The Red Wire (VCC) of all four Servo Motors (MG996R and MG90S) is connected directly to the Positive (+) of an External SMPS (5V, 10A) or heavy battery.
    - **Common Ground:** The GND of the ESP32 and the GND of the External Power Supply must be connected to each other with one wire.
- **Why (Why is it done this way?):**
    - The in-built voltage regulator of the ESP32 can handle a maximum current of 1 Ampere.
    - If we power the servos directly from the ESP32, there will be a voltage drop (Brownout) when the motors rotate, and the ESP32 will repeatedly crash/restart.
    - Common Ground is necessary so that a common Zero-Voltage Reference (0V) is established between the ESP32 control signals (PWM) and the Motors.
    - If the Ground is not common, the motors may start vibrating (jittering) abnormally.

2. Servo Motors (PWM Control)

PWM (Pulse Width Modulation) signals are sent to the servos to tell them their position.

- **The Connections: (Yellow/Orange Signal Wires)**

- Base (MG996R): GPIO 13
- Lift / Lower Arm (MG996R): GPIO 12
- Elbow / Mid Arm (MG90S): GPIO 14
- Gripper (MG90S): GPIO 27

- **Why (Why these Pins?):**
  - The ESP32 has dedicated Hardware PWM Timers.
  - These specific pins (12, 13, 14, 27) are among the ESP32's best output pins, which can continuously provide smooth, high-frequency signals without interruption.
  - This makes the movements of the robotic arm very smooth and industrial-grade.

3. Conveyor Relay Module (JQC3F-05VDC-C)

For safely turning the Heavy DC Motor ON and OFF.

- **The Logic Connections (Low Voltage Side):**
  - IN (Signal): GPIO 4 (from ESP32)
  - VCC: 5V (from ESP32's VIN pin)
  - GND: From ESP32's GND

- **The Load Connections (High Voltage Side):**
  - COM (Common): Positive (+) wire of the Motor's Battery/Power Supply.
  - NO (Normally Open): Positive (+) wire of the Conveyor Motor.
  - The Motor's Negative wire is connected directly to the Battery's Negative.

- **Why (Why is it done this way?):**
  - Microcontrollers (such as the ESP32) cannot provide enough current to directly run a motor; they will burn out.
  - A Relay is an electronic switch.
  - The ESP32 sends only a light 3.3V signal from GPIO 4, which activates the electromagnet (coil) inside the relay.
  - The coil then pulls a heavy physical switch, completing the motor's circuit by connecting NO to COM.

4. TCS34725 Color Sensor (I2C Communication)

This sensor accurately detects the color of objects.

- **The Connections:**
    - VIN / VCC: 3.3V (ESP32's 3V3 Pin)
    - GND: ESP32 GND
    - SDA (Data): GPIO 21
    - SCL (Clock): GPIO 22
    - LED: Directly connected to GND
- **Why (Why is it done this way?):**
    - **I2C Protocol:** This sensor uses the I2C protocol, which can send a large amount of complex data (RGB values) very quickly using only 2 wires (SDA and SCL).
    - GPIO 21 and GPIO 22 are the Default I2C pins on the ESP32.
    - **3.3V Logic:** This sensor operates at a 3.3V logic level. Supplying it with 5V can damage its I2C chip.
    - **LED to GND:** (As we had fixed earlier) If the LED pin is left floating, it can act like an antenna and catch electromagnetic noise from the servo motors, which can cause the I2C clock line to crash.
    - Connecting it to GND completely eliminates the noise and prevents the sensor from freezing.