Electronic components, including Arduino sensors and modules, are designed to draw only the amount of current they need based on their internal circuitry and resistance.

How Sensors and Electronics Draw Current

1. **Voltage is Supplied**
    - The Arduino or power supply provides a **fixed voltage** (e.g., 5V or 3.3V).
    - The sensor or electronic component **determines how much current it needs** based on its internal resistance and circuit design.
2. **Ohm's Law Applies**
    - The current drawn depends on **Ohm's Law**:

$$I = \frac{V}{R}$$

    - If a component has a built-in resistor or internal circuit, it **limits the amount of current it takes**.
3. **Components Only Take What They Need**
    - Unlike a **short circuit**, where current flows unrestricted, sensors and modules have a **designed operating range**.
    - For example:
        - An **LED with a 220Ω resistor** at **5V** will draw about

$$I = \frac{5V}{220\Omega} = 22.7\,mA$$

        - An **MG811 CO_2 sensor** has an **internal heating element** that draws around **200mA** when powered.
4. **Arduino Digital Pins Have a Limit**
    - Arduino **pins do not "push" current**. Instead, **the connected device pulls only what it needs**.
    - Each **digital pin** can provide **up to 20mA safely**, with a **max total of 200mA for all pins combined**.

What Happens If a Sensor Needs More Current?

If a sensor or module **requires more current than an Arduino pin can provide**, you need to:

- **Use an external power source (like a separate 5V supply)**
- **Use a transistor or MOSFET as a switch** to control high-current loads
- **Check the datasheet** to ensure the sensor is within safe limits

If you connect a sensor or any device that **draws more current than the Arduino pin can safely provide**, it **can damage or even permanently burn out the pin or the microcontroller itself**.

Why Can a High-Current Sensor Burn the Pin?

1. **Arduino Digital Pins Have a Current Limit**
    - Most Arduino boards (like the **Uno**) have a **maximum safe current per pin of 20mA** (absolute max **40mA**, but exceeding 20mA is risky).
    - The **total** current for all pins combined should not exceed **200mA**.
2. **Excess Current Can Overheat and Damage the Pin**
    - If a sensor **draws more current than the pin can handle**, the **pin's internal transistor (MOSFET) overheats**, potentially **burning out the pin permanently**.
    - Once burned out, the pin **may stop working completely** or behave unpredictably.
3. **Microcontroller Damage**
    - If a component **draws too much current**, it **can also damage the entire microcontroller (ATmega328P on an Uno)**, making the **Arduino unusable**.

Example: When a Sensor Draws Too Much Current

Bad Example: Connecting a 100mA sensor **directly to an Arduino pin**

- **The pin may overheat and get damaged**
- **The sensor might not work properly**

Good Example: Using a transistor to switch **external power**

- **Arduino controls the transistor**
- **Sensor gets full required current from an external power supply**

Key Takeaways

Sensors and electronic components **only draw the current they need**.
Arduino **provides voltage**, and the **component regulates its own current**.
High-current devices may need **external power** instead of drawing directly from the Arduino.

[Voltage, Current, Resistance, and Ohm's Law]