**Introduction**

The IoT-based water management system is designed to **automate water control** in a tank and a tap using an ESP32 microcontroller. It ensures that the water pump is turned on when the tank is empty and off when full. Additionally, a **water flow sensor** monitors tap overflow and controls a **solenoid valve** to prevent wastage. The system also features a **web-based control panel** for manual operation via a local IP address.

**Working Principle**

**1. Water Level Monitoring and Pump Control**

* The system uses **two water level sensors** inside the **main water tank**:
  + **Low-Level Sensor**: Detects when the tank is empty.
  + **High-Level Sensor**: Detects when the tank is full.
* When the water level **drops below the low-level sensor**, the ESP32 **activates the relay** to **turn ON the water pump**.
* When the water level **reaches the high-level sensor**, the ESP32 **deactivates the relay**, turning **OFF the pump** to prevent overflow.

**2. Tap Overflow Detection and Automatic Shutoff**

* A **water flow sensor** (YF-S201) is placed at the **tap outlet** to measure **water flow rate**.
* The ESP32 continuously monitors the **flow rate**.
* If the flow rate is **too high for an extended time** (indicating overflow), the ESP32 **activates the solenoid valve via a relay**, cutting off the water supply.
* This prevents **water wastage** and unnecessary overflow.

**3. Power Management**

* A **12V DC power supply** is used to power the system.
* A **buck converter (LM2596)** steps **12V down to 5V** to power the ESP32 and other components.
* The ESP32 operates at **3.3V logic**, but its **VIN pin** can handle **5V input** from the buck converter.
* The **relay module** and **flow sensor** require **5V**, which is supplied via the buck converter.

**4. IoT Web Interface for Manual Control**

* The ESP32 hosts a **web server** accessible via a **local IP address** (e.g., 192.168.1.X).
* The web interface allows the user to:
  + **Manually turn ON/OFF the pump**.
  + **Manually open/close the solenoid valve**.
  + **View water level and flow rate data in real-time**.
* The ESP32 processes user commands and updates the relay states accordingly.

**Conclusion**

This **IoT-based water management system** automates water control using **sensors, relays, and an ESP32 microcontroller**. It prevents water wastage by detecting **tank levels** and **tap overflow** while providing **remote control via a web interface**. The use of a **buck converter ensures stable power distribution**, making the system efficient and cost-effective.