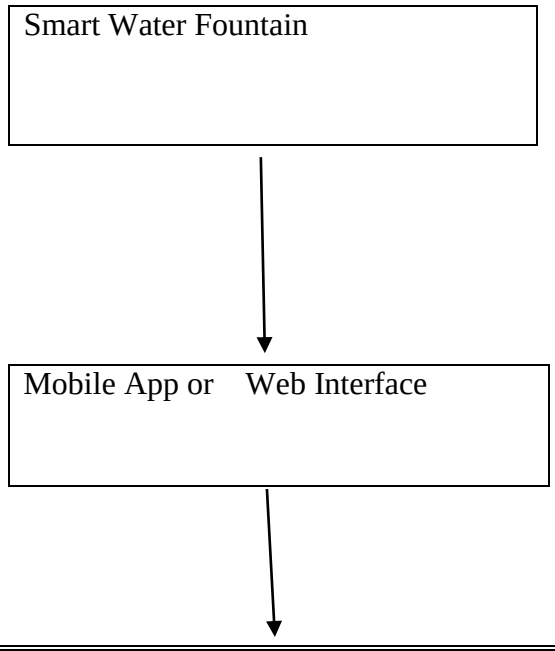
PROGRAM FOR SMART WATER FOUNTAIN:

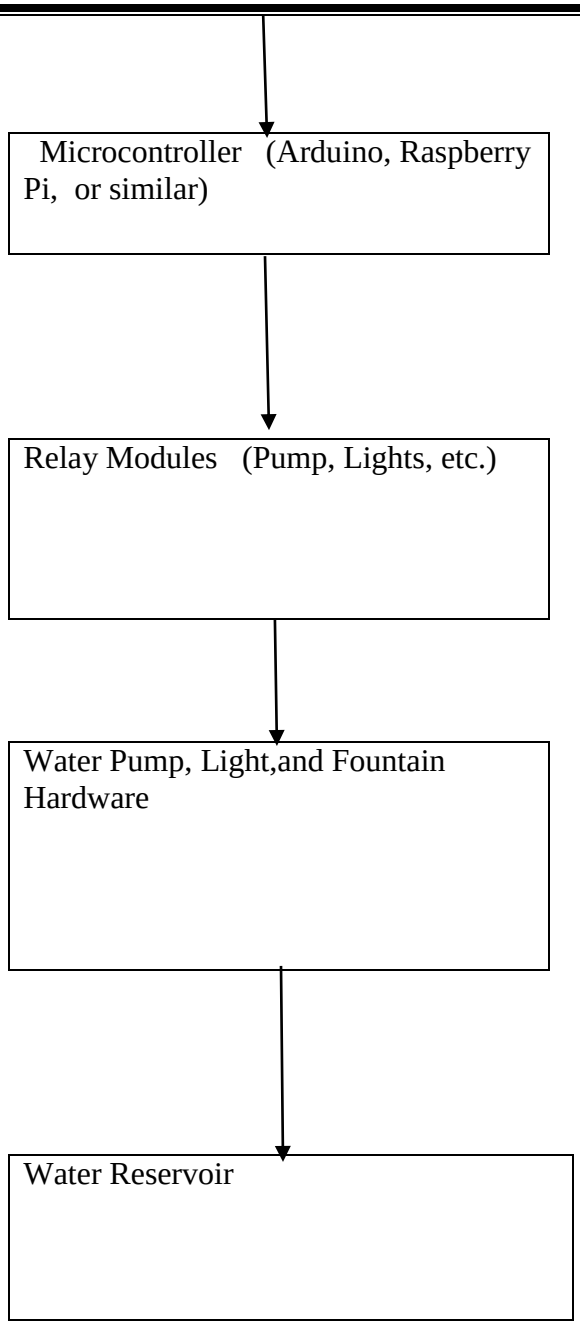
```
#include <Wire.h>
#include <WiFi.h>

// Define your network credentials
const char* ssid = "YourWiFiNetwork";
const char* password = "YourWiFiPassword";

// Define relay pins for controlling the pump and lights
const int pumpRelayPin = 2;
const int lightRelayPin = 3;

// Define a variable to track the fountain status
bool fountainOn = false;

void setup() {
  // Initialize serial communication
  Serial.begin(115200);

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
```

```
  }
  Serial.println("Connected to WiFi");

  // Set relay pins as outputs
  pinMode(pumpRelayPin, OUTPUT);
  pinMode(lightRelayPin, OUTPUT);

  // Turn off the pump and lights initially
  digitalWrite(pumpRelayPin, LOW);
  digitalWrite(lightRelayPin, LOW);
}

void loop() {
  // Check for commands from a remote control source (e.g., a mobile app or web interface)
  // Implement your communication protocol here

  // Sample logic to control the fountain
  if (fountainOn) {
    turnFountainOff();
  } else {
    turnFountainOn();
  }

  // Add any other logic or sensor reading here
}

void turnFountainOn() {
  digitalWrite(pumpRelayPin, HIGH);
```

```
  digitalWrite(lightRelayPin, HIGH);
  fountainOn = true;
  Serial.println("Fountain turned on");
}

void turnFountainOff() {
  digitalWrite(pumpRelayPin, LOW);
  digitalWrite(lightRelayPin, LOW);
  fountainOn = false;
  Serial.println("Fountain turned off");
}
```

BLOCK DIAGRAM FOR SMART WATER FOUNTAINS :





DATASET OF SMART WATER FOUNTAINS :

Fountain Operation Data:

Timestamp:

Date and time of data collection.

Fountain status:

On or off.

Pump flow rate:

Flow rate of the water pump.

Light settings:

Control data for fountain lighting (e.g., color, intensity).

Water Quality Data:

Water level:

Measured water level in the fountain reservoir.

Water temperature:

The temperature of the water.

pH level:

The pH level of the water.

Electrical Conductivity (EC): Measures of water's ability to conduct electrical current.

Environmental Data:

Ambient temperature:

Temperature of the surrounding environment.

Humidity:

Humidity levels in the vicinity of the fountain.

Light intensity:

Ambient light conditions, which could affect fountain lighting.

Weather conditions:

Rain, wind, etc. (if the fountain is outdoors).

Energy Consumption Data:

Power consumption:

Energy consumption of the fountain's components.

Energy usage patterns:

How energy usage varies with the fountain's operation.

User Interaction Data:

User commands:

Data indicating when a user turned the fountain on/off or changed settings.

User preferences:

User-defined settings or preferences for the fountain's operation.

Error and Alert Data:

System errors:

Logs of errors or issues that occurred during operation.

Alert messages:

Notifications or alerts sent to users or administrators.

Maintenance Data:

Maintenance schedule:

Records of routine maintenance activities.

Component health:

Data on the condition and performance of fountain components (e.g., pump condition, light bulb lifespan).

Remote Control and App Usage Data:

Mobile app or web interface usage:

Data on how often users interact with the control interface.

User authentication data:

If user accounts are involved, information about user logins and actions.