1. Soil Moisture Reading Logic

- The soil moisture sensor outputs an analog voltage.
- The Raspberry Pi Pico reads this using its **ADC pin (GP26)**.
- The raw ADC value is converted into a **percentage moisture value**.

Logic:

1. Read analog value → raw = soil.read_u16()
2. Convert to moisture % using inverse mapping
   (because dry soil = high ADC value)

   ```
   percent = int((65535 - raw) * 100 / 65535)
   ```

3. Return moisture percentage for further decisions.

---

2. DHT22 Temperature & Humidity Logic

- The DHT22 sensor is initialized on **GP17**.
- Temperature and humidity are read using:

```
dht_sensor.measure()

temp = dht_sensor.temperature()

hum = dht_sensor.humidity()
```

- Errors during sensor reading are handled using try/except to ensure loop stability.

---

3. Water Pump Control Logic

The pump relay is connected to **GP16**.

Decision Logic:

- If moisture is **below the threshold**, soil is dry → Pump must run.
- If moisture is **above the threshold**, soil is wet → Pump must stop.

```
if moisture < MOISTURE_THRESHOLD:

   pump.value(1)   # Pump ON

else:

   pump.value(0)   # Pump OFF
```

- Status (1 = ON, 0 = OFF) is stored for ThingSpeak updates.

---

4. Fan Control Logic

Fan relay is connected to **GP15**.

Decision Logic:

- If temperature is **greater than or equal to the set limit**, fan is turned ON.
- Otherwise, fan remains OFF.

```
if temp >= TEMP_THRESHOLD:
  fan.value(1)  # Fan ON
else:
  fan.value(0)  # Fan OFF
```

- Fan status is also logged for cloud monitoring.

---

5. Cloud Upload Logic (ThingSpeak)

Sensor readings and device statuses are uploaded to the cloud every cycle.

Steps:

1. Build the ThingSpeak URL with:
    - Temperature
    - Humidity
    - Moisture
    - Pump status
    - Fan status

---

6. Continuous Monitoring Loop

All functions run inside an infinite loop:

Loop Operations:

1. Read soil moisture
2. Read DHT22
3. Print all values
4. Apply pump control logic
5. Apply fan control logic
6. Upload data to ThingSpeak
7. Delay for 1 second