

**Flow Chart Description**

**1. START**

* Begin the system operation.

**2. Initialize Hardware and Connectivity**

* Initialize:
  + **SHT20 sensor** (for temperature and humidity)
  + **GPS module with GPRS** (used for MQTT connectivity)
  + **LED tower** (with 4 color indicators: Green, Blue, Orange, Red)
  + **MQTT client**
* Establish serial communication and configure pins.

**3. Connected to MQTT?**

* System checks if the MQTT connection (through GPS/GPRS) is active.
* **If NO:**
  + Reattempt connection.
  + Reinitialize if required.
* **If YES:**
  + Continue to next steps.

**4. Get Threshold Values**

* Load or confirm the following threshold heat index levels:

| **Level** | **Heat Index (°C)** | **LED Color** | **Status** |
| --- | --- | --- | --- |
| Normal | < 27 | 🟢 Green | Safe |
| Warning | 27 – 33 | 🔵 Blue | Mild Heat |
| Alert | 34 – 42 | 🟠 Orange | High Heat |
| Critical | > 42 | 🔴 Red | Dangerous |
| No reading / Disconnected | — | ⚫ Off | No data |

**5. Get Heat Index**

* Read **temperature** and **humidity** from the **SHT20**.
* Calculate **heat index** using the standard formula (converted for Celsius):

**6. Send Data to Web (via MQTT)**

* Publish the following data to your MQTT topic:
  + Temperature (°C)
  + Humidity (%)
  + Heat Index (°C)
  + Threshold Status (Normal, Warning, Alert, Critical)

**7. Threshold Trigger?**

* Compare the current **heat index** against thresholds:
  + Decide the current **status** and **LED color**.
  + Trigger corresponding actions.

**8. LED Tower Indication**

| **Condition** | **Action** |
| --- | --- |
| HI < 27 | Turn **Green LED ON** |
| 27 ≤ HI < 33 | Turn **Blue LED ON** |
| 34 ≤ HI < 42 | Turn **Orange LED ON** |
| HI ≥ 42 | Turn **Red LED ON** |
| No reading / Error | Turn **ALL OFF** |

Only one LED should light at a time.

### ****9. Send SMS (Optional for Alert or Critical)****

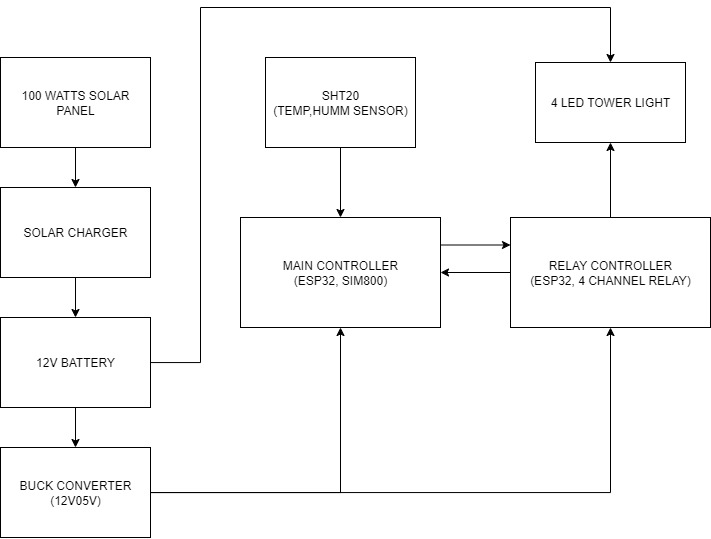
| **Condition / Heat Index Level** | **Example Message** |
| --- | --- |
| **HI < 27°C — NORMAL** | 🟢 “Heat Index 27°C. Presko pa! Ligtas sa labas pero tubig-tubig din para iwas uhaw.” |
| **27°C ≤ HI < 33°C — WARNING** | 🔵 “Heat Index 33°C. Medyo mainit na! Iwas bilad, magpayong at uminom ng maraming tubig.” |
| **34°C ≤ HI < 42°C — ALERT** | 🟠 “Heat Index 42°C! Grabe ang init, iwas muna sa labas 10AM–4PM. Stay cool at hydrated!” |
| **HI ≥ 52°C — CRITICAL** | 🔴 “Heat Index 52°C! Delikado na, huwag nang lumabas. Uminom ng tubig at manatili sa malamig na lugar.” |
| **No reading / Error** | ⚫ No SMS sent. |

**10. Wait Interval**

* Pause for a fixed period (3–5 minutes) before the next reading cycle.
* Helps reduce network traffic and power consumption.

**11. Loop**

* Repeat from “Get Heat Index” continuously for real-time monitoring.



**1. 100 Watts Solar Panel**

* **Function:** Converts sunlight into DC electricity.
* **Output:** Typically 18V DC (open circuit) / 12V nominal.
* **Purpose:** Provides renewable power to charge the battery through the solar charger.

**2. Solar Charger**

* **Function:** Regulates and manages the charging of the **12V battery** from the solar panel.
* **Key Features:**
  + Prevents overcharging or deep discharge.
  + Distributes power efficiently to the system.

**3. 12V Battery**

* **Function:** Stores energy from the solar panel.
* **Purpose:** Powers the system continuously, even at night or during cloudy conditions.
* **Connection:** Supplies power to the buck converter and other components.

**4. Buck Converter (12V → 5V)**

* **Function:** Steps down the 12V battery output to 5V.
* **Purpose:** Provides stable **5V DC power** for the **ESP32**, **SHT20**, **SIM800**, and **relay modules**.

**5. Main Controller (ESP32 + SIM800)**

* **Core of the system.**
* **Functions:**
  + Reads **temperature** and **humidity** from the **SHT20 sensor**.
  + Calculates **Heat Index**.
  + Sends sensor data to **MQTT server** via **SIM800 (GPRS)**.
  + Determines the **threshold level** (Normal, Warning, Alert, Critical).
  + Sends control commands to the **Relay Controller** for LED light indication.
  + Optionally sends **SMS alerts** when critical conditions are reached.

**6. SHT20 (Temperature & Humidity Sensor)**

* **Function:** Measures the ambient **temperature (°C)** and **humidity (%)**.
* **Communication:** I²C connection to the ESP32 main controller.
* **Purpose:** Provides real-time environmental readings for heat index calculation.

**7. Relay Controller (ESP32 + 4-Channel Relay)**

* **Function:** Controls the **4-color LED Tower Light**.
* **Input:** Receives control signals from the **main controller**.
* **Output:** Activates one relay at a time, turning ON the LED corresponding to the current heat index level.
  + Relay 1 → Green (Normal)
  + Relay 2 → Blue (Warning)
  + Relay 3 → Orange (Alert)
  + Relay 4 → Red (Critical)

**8. 4-Color LED Tower Light**

* **Visual Indicator** showing real-time heat index condition:

| **Heat Index Range** | **Color** | **Meaning** |
| --- | --- | --- |
| < 27°C | 🟢 Green | Normal |
| 27–33°C | 🔵 Blue | Warning |
| 34–42°C | 🟠 Orange | Alert |
| ≥ 52°C | 🔴 Red | Critical |

**System Flow Summary**

1. **Power Source:**  
   Solar panel → Solar charger → 12V battery → Buck converter (12V→5V)
2. **Sensor Reading:**  
   SHT20 → Main Controller (ESP32)
3. **Data Processing:**  
   ESP32 calculates **heat index** and determines threshold level.
4. **Communication:**  
   ESP32 (via SIM800) sends data to **MQTT server** or via **SMS** when needed.
5. **LED Indication:**  
   ESP32 sends signal → Relay Controller → Activates correct LED color.
6. **Continuous Operation:**  
   Powered 24/7 by solar and battery backup.

### ****Power Efficiency Notes****

* The **100W Solar Panel** provides sufficient energy during daylight to:
  + Power the **entire system** in real-time, and
  + **Recharge the 12V battery** simultaneously for nighttime operation.
* **Typical Power Consumption:**
  + **ESP32** – ≈150 mA
  + **SIM800 (GPRS Transmission)** – ≈2 A (peak during TX)
  + **Relays and LED Tower Light** – ≈200–500 mA total
  + **SHT20 Sensor** – <10 mA
  + **Solar Charger and Buck Converter losses** – ≈10–15% total overhead
* The **12V 20Ah battery** (typical) stores enough energy to:
  + **Power the full system for approximately 12–18 hours** without sunlight,
  + **Ensure continuous operation** during night or cloudy conditions.
* The **solar charger** automatically:
  + Distributes power between load and battery,
  + Prevents overcharge or deep discharge,
  + Ensures stable and efficient energy management.

**Assumptions (explicit)**

* **Panel:** 100 W nominal.
* **Average effective sun (peak-sun hours):** **4.5 h/day** (reasonable for PH).
* **System/component consumption (typical estimates):**
  + ESP32 ≈ **150 mA @ 5 V** → 5 × 0.150 = **0.75 W**
  + SIM800 average ≈ **1.0 W** (peaks to ~2 A during TX but average much lower if you batch transmissions)
  + SHT20 ≈ **0.05 W**
  + Relays + LED tower (typical usage) ≈ **0.5 W** (only one LED on at a time; not always 100% duty)
* **Power conversion & charger losses:** **15%** total overhead.
* **Battery:** 12 V × 20 Ah = **240 Wh** nominal. Useable assumed at **50% DoD → 120 Wh** usable.

**Solar energy available per day**

Panel energy (raw):  
100 W × 4.5 h = **450 Wh/day**

Account for derating (wiring, angle, dust, temp) — I used **75% system derate**:  
450 Wh × 0.75 = **337.5 Wh/day** available to charge + run loads.

**Typical (realistic) load calculation**

Sum the power before losses:

* ESP32 = 0.75 W
* SIM800 (avg) = 1.00 W
* SHT20 = 0.05 W
* LED/relays (typical) = 0.50 W  
  Total = 0.75 + 1.00 + 0.05 + 0.50 = **2.30 W**

Account for 15% system losses by dividing by (1 − 0.15):  
Average system power = 2.30 / 0.85 = **≈ 2.71 W**

Daily energy use = 2.71 W × 24 h = **≈ 64.9 Wh/day**  
Night energy (hours without sun ≈ 24 − 4.5 = 19.5 h) = 2.71 W × 19.5 h = **≈ 52.8 Wh**

**Compare with solar & battery:**

* Solar provides **337.5 Wh/day** → easily covers **64.9 Wh/day**.
* Night energy **52.8 Wh** is less than battery usable **120 Wh** → battery covers the night with margin.

**Conclusion (typical):** 100 W panel + 12 V 20 Ah battery **can support 24/7 operation easily** in normal use.