

Power and Component Audit for GadgetBuddy

This document outlines the power consumption and component list for the GadgetBuddy project, providing a critical reference for anyone continuing development, especially when planning for power sources and battery life. It includes power calculations for two possible microcontrollers: the Arduino Uno and the ESP32.

Project Hardware Overview

The device is composed of the following components, all operating at a standard 5V input voltage unless otherwise noted.

- **Microcontroller:** Arduino Uno or ESP32
- **Display:** LCD I2C 20x4
- **Sensors:** MQ135 (Gas), DHT11 (Temp/Humidity), GY-271: QMC5883L (Compass)
- **Audio:** TEA5767 (FM Radio), GF1002-PAM8403 (Audio Amplifier)
- **Interface:** 2 standard buttons, 1 simple RGB LED
- **Clock:** RTC Clock DS1307

Power Consumption Analysis

The following table provides the estimated current draw for each component. These figures are used to calculate the overall power requirements for each microcontroller option. Note that certain components, like the audio amplifier, have variable consumption that depends on their usage.

| Component | Typical Current Draw | Notes |
|------------------|----------------------|--|
| Arduino Uno | ≈45mA | Baseline current for the board. |
| ESP32 | ≈80mA | Baseline current, including power for the Wi-Fi/Bluetooth module (in a typical, active state). |
| LCD I2C 20x4 | ≈25mA | Includes the backlight. |
| MQ135 Gas Sensor | ≈150mA | The heater element is the primary power consumer. |
| DHT11 Sensor | <1mA | Very low power, negligible for total calculation. |
| TEA5767 FM Radio | ≈15mA | Active mode current. |
| GF1002-PAM8403 | ≈150mA | Assumed average draw for moderate volume. |
| RGB LED | ≈20mA | Assumed one color is on at a time. |
| RTC Clock DS1307 | ≈1.5mA | Low power, for timekeeping. |

| | | |
|--------------------|------|--|
| GY-271: QMC5883L | ≈2mA | Low power for the magnetometer module. |
| 2 Standard Buttons | 0mA | No current draw. |

Total Power Requirements and Battery Capacity

The total current draw varies significantly depending on the microcontroller chosen. The calculations below provide the required battery capacity in milliampere-hours (mAh) for both a 5-hour and 8-hour duration.

Arduino Uno

Using the Arduino Uno, the total estimated current draw is approximately **410 mA**.

- **5-hour duration:** A minimum battery capacity of **2050 mAh** is required.
- **8-hour duration:** A minimum battery capacity of **3280 mAh** is required.

ESP32

Using the ESP32, the total estimated current draw is approximately **445 mA**.

- **5-hour duration:** A minimum battery capacity of **2225 mAh** is required.
- **8-hour duration:** A minimum battery capacity of **3560 mAh** is required.

Note: It's a good practice to select a battery with a capacity slightly higher than the minimum required to account for battery degradation and unexpected power spikes.

Future Considerations

When making changes to the device, consider the following:

- **Power-Saving Modes:** The ESP32 offers advanced sleep modes (e.g., deep sleep) that can dramatically reduce power consumption when the device is idle. This would be a crucial feature to implement for maximum battery life.
- **Component Selection:** Every new component will add to the overall current draw. Be sure to check its power consumption and adjust the battery requirements accordingly.
- **Voltage Regulation:** Ensure a stable 5V supply is provided to all components.

This document should serve as a helpful reference for anyone working on this project. For a more detailed breakdown, refer to the individual component datasheets.