Particle

Power Management for IoT Devices

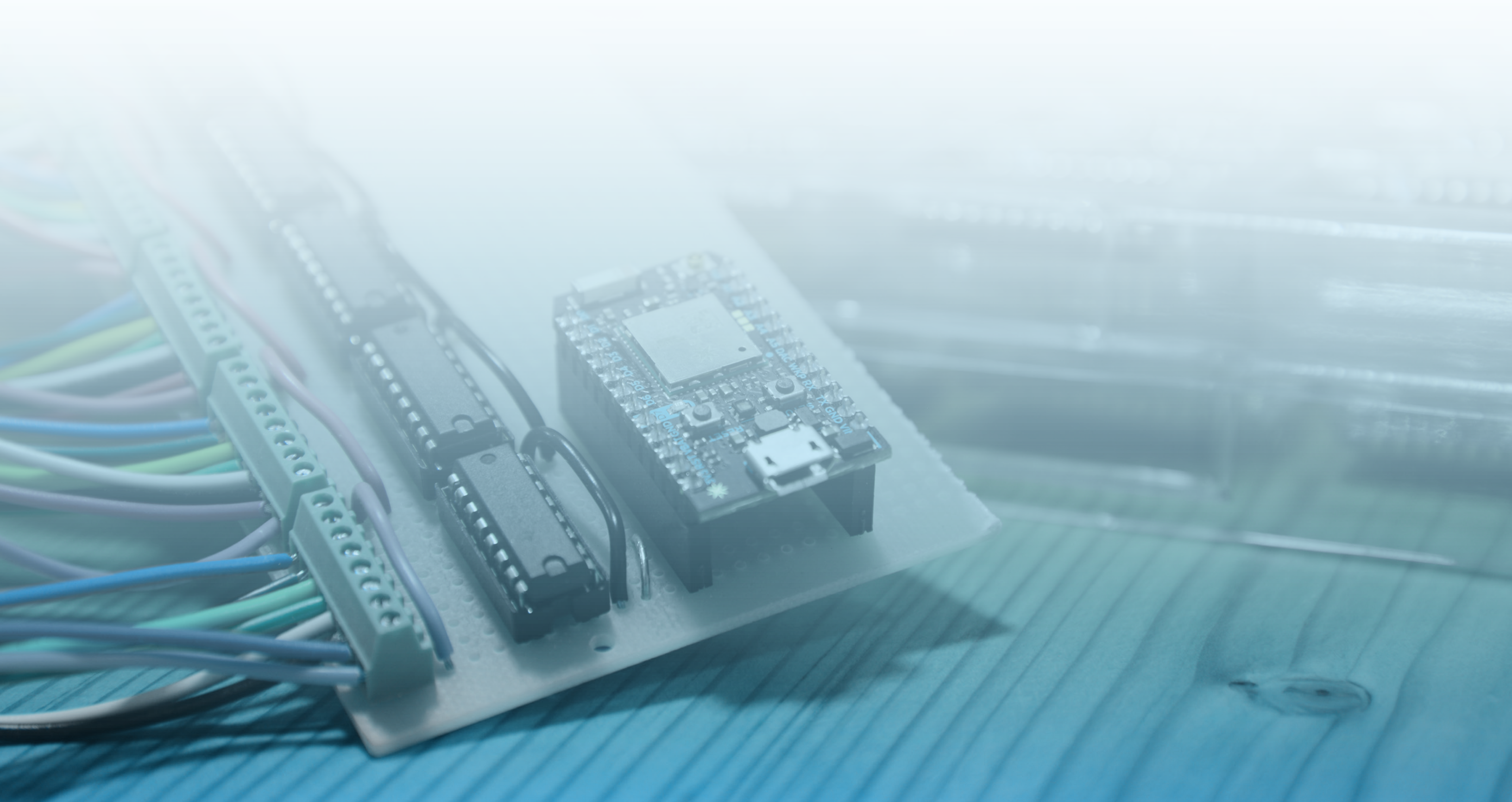


Table of Contents

Introduction (3)

Energy Budget Modeling (4)

Reducing Power Consumption (7)

Power Storage (12)

Power Generation (15)

Introduction

Developing an IoT product is an iterative design process. At every stage of the process — from prototype to production — power consumption, storage, and generation must be considered. Often, IoT devices are deployed remotely and are not connected to the power grid. For these kinds of devices, using best design practices to reduce power consumption will help to lengthen the lifetime of the product and extend its value to the customer.

This paper will provide an overview of the following topics:

- **Energy Budget:** Creating a model for IoT device power use
- **Reducing Consumption:** Best practices to minimize the power consumption of IoT devices
- **Power Storage:** Important information for selecting a battery for IoT devices
- **Power Generation:** Methods of generating power and extending the battery life of IoT devices

Energy Budget Modeling

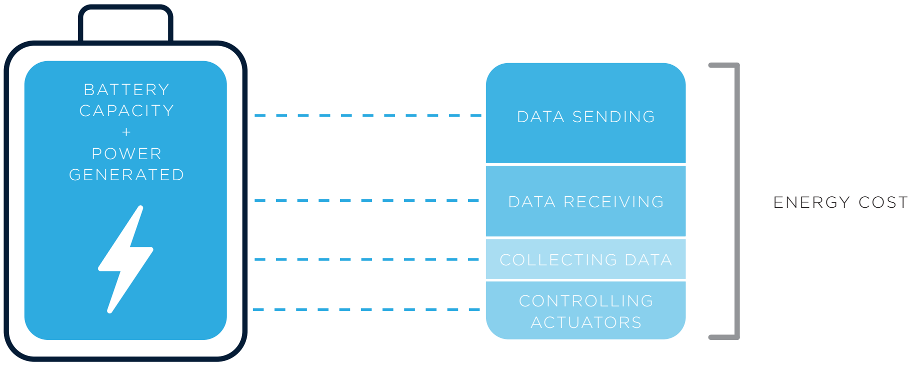
Most remotely deployed IoT devices have a finite amount of power available to them that limits the amount of time that they can operate in the field before they need to be recharged or replaced. The behavior of the device is as important to its operational lifetime as the battery that powers it, so it is important to develop an energy budget model for an IoT device in development.

IoT devices have certain components that use power while operating - these include:

- A radio that allows the device to communicate
- A microprocessor or microcontroller (MCU)
- Peripheral components such as sensors and actuators

Generally, the energy budget model for most IoT devices can be described by the following equation:

Power remaining = (battery capacity) – (send data) – (receive data) – (collect data) – (control actuators) – (ambient loss) + (power generated)



Battery Capacity

Most remote IoT devices depend on a battery as a primary source of power. Battery capacity is typically measured in milliamp-hours (mAh) and this parameter should be readily available for any limited power source. Because batteries are typically the primary power source for remotely deployed devices, the capacity of the battery will have a direct correlation to its duration of use. There are many sizes and chemistries of battery to consider, but one primary distinction is between rechargeable batteries like lithium-polymer (Li-Po) batteries and non-rechargeable batteries like alkaline batteries.

Sending and Receiving Data

Most IoT devices have a wireless radio that they use to send and receive messages, and for most devices, these actions have the highest instantaneous cost to the device's energy budget. For most wireless protocols including Wi-Fi, cellular (2G/3G/LTE), and Bluetooth, it is a more costly action to send data than it is to receive data. The power consumption to both send (Tx) and receive (Rx) modes should be available in the datasheet for the wireless radio module.

There are a variety of reasons IoT devices send and receive data. For example, devices can upload sensor data to online databases, send messages to other IoT devices, or update the device's status online. They can receive data to update the firmware running on their devices, take commands from the device's owner, or retransmit data from other IoT devices to the internet.

Collecting Data from Sensors

Many IoT devices read environmental data from sensors connected to the device in order to make that information accessible remotely or to trigger a resulting action. Different kinds of sensors require different amounts of power, so considering power consumption alongside cost, accuracy, size, and other variables is important when designing an IoT product.

Controlling Actuators

An actuator is a general term for a component of an IoT device that is responsible for moving or controlling another mechanism or system. An example of an actuator is the motor that unlocks the deadbolt on a connected door lock or a solenoid that toggles a light switch from on to off. Like with wireless communication and collecting sensor data, the impact than an actuator has on the device's energy budget will be determined by the current demands of the action and how often the action occurs.

Ambient Current Consumption

The ambient current consumption of an IoT device is the rate that it consumes power when it is not acting in any of the ways described above. This ambient current consumption rate is a composite value that includes the steady state current consumption of the microcontroller, wireless radio, and the rest of the electrical system.

It is possible to significantly reduce the ambient current consumption of an IoT device by leveraging sleep modes available to the device radio and microcontroller. Strategies for reducing ambient current consumption are described in the next section, **Consumption**.

Calculating Cost to Energy Budget

It is possible to model the power consumed for a particular action (receiving a message, in the example below) using the following framework:

*Power consumed = (current consumption) [mA] * (duration of transmission) [s] / 3600*

So, for a Wi-Fi device receiving a firmware update over a period of 10s, the estimated power consumption is

*Power consumed = 350mA * 10s / (3600 s/hr) = 0.97mAh*

Reducing Power Consumption

Reducing power consumption of IoT devices that are deployed remotely or have a constrained power source has several benefits. If the device is battery powered, the run time between charges can be increased. If the device is solar-powered, the size and cost of the panel can be reduced.

Each major consumer of power (radio, micocontroller, sensors, actuators) can be optimized within the design of the IoT device to consume less power.

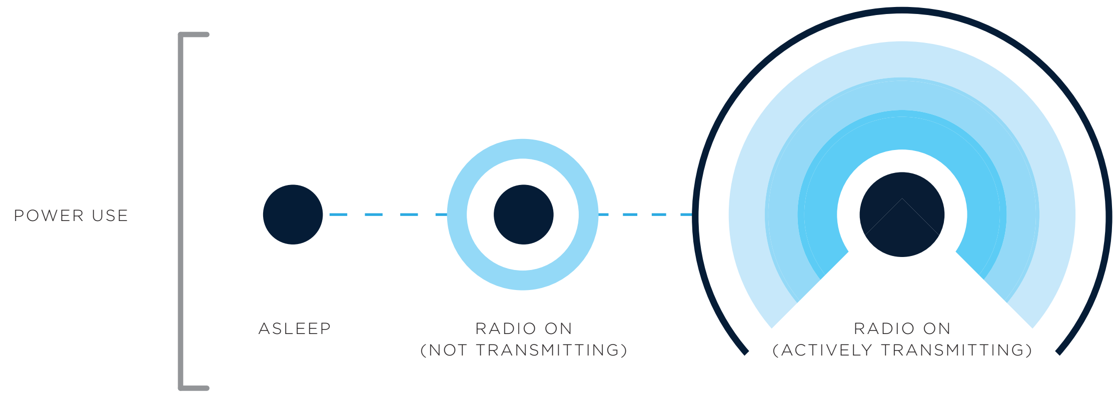
Wireless Radios

When designing an IoT device, a product creator must carefully select the type of connectivity. Radios have trade-offs in terms of range and power usage, generally illustrated by the chart below:

	Cellular (2G/3G/LTE)	**Wi-Fi**	**BLE**
Power consumption (Tx)	700 mA	300 mA	18 mA
Range	Many miles	100-300 ft.	30 ft.
Applications	Mobile, remote	Smart home	Wearables

Most product creators will choose a radio based on the needs of the application. Bluetooth radios are great for small, low-powered devices, but need a gateway to communicate with the internet. Wi-Fi devices require a Wi-Fi network and router to communicate. Cellular devices are freed from the constraints of Wi-Fi, enabling much more remote applications. As you move from Bluetooth to cellular, the power required to send and receive messages increases accordingly.

Leveraging Low Power Modes to Reduce Energy Consumption



The best way to conserve power for an IoT device with a wireless radio is to ensure that the radio is only fully powered when actively in use. Whenever possible, the radio should be powered off when not in use. If the device is collecting and transmitting sensor data, it is preferable to keep the radio turned off until needed, storing data for transmission in the interim.

If the radio must be on in a Wi-Fi device, it is possible to keep it in a more efficient "power saving mode" which sets the internal oscillator to a lower frequency to conserve power. This can only be done when the radio is not actively transmitting, but can result in significant power savings and decreased impact to the device's energy budget.

Grouping Messages to Reduce Bandwidth

Every byte of data transmitted lengthens the amount of time the radio must be fully powered. Additionally every separate message transmitted requires some amount of data overhead and thus power usage overhead. Both of these factors are within a product creator's control.

Rather than sending every sensor reading immediately to the network, a product creator can collect a reading every 5 minutes, save it locally on the device, and send the list of collected readings once every 2 hours. Thus the same amount of information can be gathered remotely with much lower power consumption. The primary trade-off in this example is latency — your readings may be up to 2 hours old.

Further savings can be obtained through data compression, which can be simple. As an example, a product creator can send the average of those readings rather than the entire list of readings. For data more complexly structured than simple numbers, a text description can be base64 encoded rather than sending the full text content.

Additional Strategies for Wi-Fi Applications:

Wi-Fi power consumption is directly correlated to the amount of data the radio is transmitting. High bandwidth applications, such as downloading web pages or streaming music, will consume much more power than low bandwidth applications like transmitting sensor data. For high bandwidth applications, it is highly recommended that the device be connected directly to an AC power supply or have a charge management circuit that allows for frequent recharging of the device.

Additional Strategies for Cellular Applications:

Cellular (2G/3G/LTE) is very similar to Wi-Fi in that power consumption is directly correlated to data usage. To reduce cellular power consumption, it is important to choose an efficient communication protocol that introduces minimum overhead. Particle has developed a low-bandwidth communication protocol for cellular applications that is 60x more efficient than a typical HTTPS request.

Leveraging dedicated API endpoints in the cloud can also conserve data usage. Instead of programming an IoT device to send and receive messages directly to and from open Internet services (ie: weather.com or AWS IoT), it is preferable to send smaller amounts of data using a byte-efficient communication protocol designed for IoT to the Cloud, where the data can be stored and processed.

An example of the bandwidth efficiencies enabled by a dedicated cloud service is a remote weather station that transmits data to multiple endpoints like weather.com, Google Cloud Platform, and NASA. Using Particle Cloud, the device only needs to consume power to transmit the data once, and retransmission to all other endpoints can be facilitated with webhooks and custom integrations.

Microcontrollers (MCUs)

Microcontrollers are the brain of most IoT devices. As such, they consume a large fraction of the energy budget in normal operation. They are similar to wireless radios in that they require a larger amount of power when they are turned on, but can be placed into sleep modes to conserve power. The largest amount of power is consumed when microcontrollers are actively running firmware.

To conserve power, it is important to fully activate the microcontroller only when the processor is needed. For example, if a product creator wanted to communicate data from a temperature sensor that samples room temperature every 10 minutes, the microcontroller should only be turned on when actively reading and transmitting sensor data. At all other times, it can be asleep.

Increasing Efficiency Using Particle Sleep Modes

The simplest way to increase microcontroller efficiency is to put it to sleep when not in use. The Particle firmware API provides simple and powerful functions for configuring devices into a variety of low power sleep modes. The table below illustrates the Particle firmware API commands that can be used to independently control the state of the microcontroller and radio to reduce power consumption.

	Radio on	**Radio off**
MCU on	Normal operation	WiFi.off() or Cellular.off()
MCU off	System.sleep() in network standby mode	System.sleep() in standard mode System.sleep() in deep sleep mode

WiFi.off() and ***Cellular.off()*** can be used to temporarily deactivate the radio, which is typically the largest consumer of power. The radio can be turned back on by the microcontroller at any time, including after a predefined period (e.g. once every hour, to send collected data), or when an external stimuli is detected (e.g. the room temperature rising above a specified threshold).

System.sleep() can be used to turn off the wireless radio, like with WiFi.off() and Cellular.off(), but also puts the microcontroller into one of three sleep modes that use a tiny fraction of the energy of normal operation.

- **Network standby mode** puts the microcontroller to sleep while keeping the radio turned on so that it can still receive messages from the cloud. Of the three System.sleep() modes described here, network standby mode uses the most power.
- **Sleep stop mode** deactivates the radio and leaves the micro controller in a “paused” state. Although the MCU is not running code or actively processing commands, power is retained to its volatile memory which allows the device to resume executing code at the place where it left off and wake up via an external interrupt to most of the exposed pins.
- **Deep sleep mode** minimizes power consumption by putting both the MCU and wireless radio to sleep. Deep sleep mode allows remotely deployed devices to last for months and on a single charge of battery, and is a very useful tool for reducing power consumption for IoT devices.

System.sleep() can be used with parameters that specify that the microcontroller wakeup after a certain period of time (60 seconds, for example) or upon the occurrence of a specified trigger event (activity on an accelerometer, for example) that activates the wakeup sequence of the device.

Sensors and Actuators

Selection of sensors and actuators can greatly impact power consumption on a device. Even within a specific type of sensor or actuator, power consumption can vary by orders of magnitude depending on model or manufacturer.

For example, a product creator might need to select an accelerometer for their device. By looking at the datasheets for Analog Devices accelerometers, large differences in power consumption are apparent. In sleep mode the ADXL193 pulls 1.5mA, but the ADXL362 pulls 10nA, a difference of 5 orders of magnitude!

In order to understand the impact that a sensor or actuator will have on a device's energy budget, it is important to refer to the component's datasheet.

Power Storage

Considerations When Selecting a Battery

There are several considerations that are important to consider when selecting a battery for IoT devices. Battery selections are generally constrained by a few factors — namely dimensional constraints imposed by use case, energy capacity constraints imposed by the required energy budget, and environmental constraints imposed by the operating conditions of the product.

Here is a checklist of questions that should be answered before selecting a battery for an IoT device:

Physical Constraints:

- How large is the enclosure for the device? Will the battery fit in the enclosure with all of the other components?
- If the battery is rechargeable, what sort of charging will the battery require? If it can be charged with an external power supply, is the circuit accessible to an external plug/adapter? Where is the charging circuit located?
- Is the weight of the product important to its performance? How much can this battery weigh? Does energy density matter (ie: capacity/weight)?

Capacity Constraints:

- How often can the battery be charged? For how long?
- Given the energy budget for the device, will the battery last between charges?

Environmental Constraints:

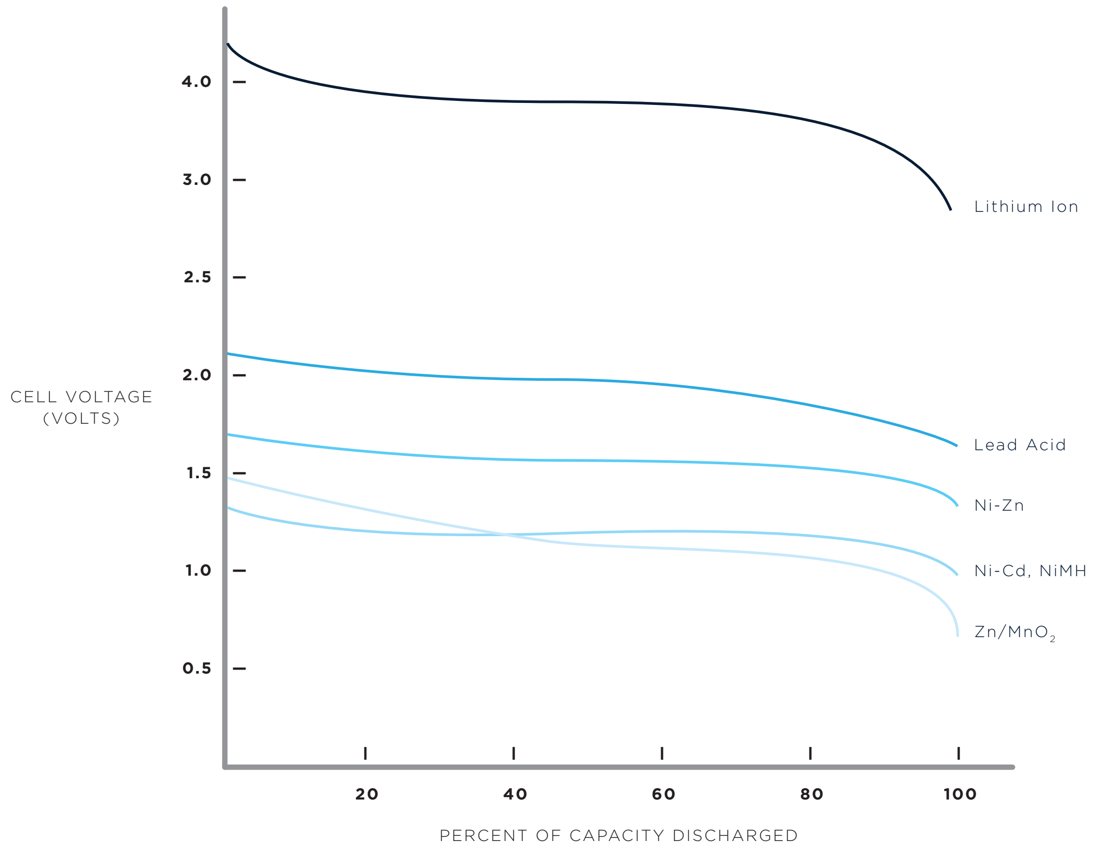
- What sort of temperature ranges will this battery be operating under? Can the power supply reliably operate in this temperature range?
- How often will the battery be replaced? Is the device deployed in a remote location? Will it last through enough charging cycles to make it to the next replacement?

Another Consideration: Battery Discharge Curves

Interpreting the capacity of a battery for energy budgeting can be difficult. Radios and microcontrollers have minimum voltages below which they will brown out or turn off altogether, and as batteries discharge the voltage they provide decreases to below their nominal voltage. A battery-powered device must be designed with the interaction between these two parameters in mind.

In specific terms, an alkaline AA battery with a rated capacity of 2000mAh and a nominal cell voltage of 1.5V could be providing only 1.0V after 1500mAh have been used. If a hypothetical device was using 3xAA in series, expecting a resulting voltage of 4.5V (3*1.5V) and requiring a minimum of 3.3V (typical of digital electronics), it would fail when the batteries still had 25%+ of their rated capacity remaining.

This can be compensated for by selecting battery chemistries and configurations with voltages much higher than minimum voltage requirement and using efficient regulators to produce the desired voltage. To learn more, look up battery discharge curves for different battery chemistries. Each chemistry (LiPo, alkaline, LiFePO4, NiMH, etc) has different curves and they are highly affected by the rate at which they are discharged.



Sample Discharge Curve for Common Battery Chemistries

Battery Safety

Batteries can be dangerous and must be handled with intelligent consideration. A battery that lacks the necessary charge and discharge protections can shorten the lifetime of the battery or result in overheating and explosion. Batteries can also explode when exposed to temperatures or voltages outside of their specified operating ranges or when they are overly confined in tight spaces. Space constraints that caused a short-circuit were recently found to be the cause of Samsung Galaxy Note 7 explosions.

It is highly recommended that any battery that is included in the electrical design of an IoT system be tested at load and certified for compliance by the appropriate regulatory body.

Particle's lithium polymer batteries include overcharge current protection and are rated for discharge at high amperages and are selected for cellular based applications. Documentation of our battery certifications can be found in our documentation.

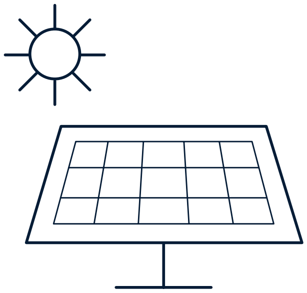
Power Generation

Types of Power Generation

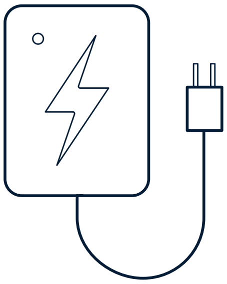
For some IoT products, it may be necessary to recharge the battery or supplement the power supplied by the battery with external energy sources. There are several strategies for generating additional power for an IoT device:

- **Episodic charging** is charging that occurs on a regular or recurring interval. Smartphones, wearables, or portable devices that require daily recharging are common examples. For designs that requires frequent charging, a battery chemistry that can tolerate a high number of charge/discharge cycles is required.

- **Intermittent charging** is charging that occurs opportunistically, like solar panels or regenerative braking on vehicles. For intermittent charging, a power source and battery must be chosen with enough capacity to continously power the device. It is also important that the availability of an external power source does not result in overcharging of the battery, so a charge protection circuit is required.



- **Battery backups** are used for devices that have an alternate power supply or are hooked up to the power grid. In this scenario, the battery is only used as a backup power supply in instances where the primary power source is unavailable.



Episodic Charging

There are several methods that can be used to replenish a rechargeable battery:

Wired Charging

Most rechargeable devices have batteries that can be charged with an external power supply. This is the fastest way to replenish a battery, and requires that an external charging interface be exposed and accessible to the end user.

Wired charging is capable of the highest charging rates and should be selected for devices that require frequent charging or have very large batteries that would otherwise take a very long time to replenish with wireless charging techniques.

There are several common interfaces that are used to charge IoT devices. They include:

- **USB variants** (5V)
- **Power barrel jack** connectors (5-24V typical), which come in a variety of sizes including 2.1mm and 2.5mm diameters
- **Ethernet** or RJ-45 connectors (48V typical), for devices that support Power over Ethernet, or PoE

Common examples of wired charging apparatuses are:

- Cell phones that are charged via 5V micro USB, USB-C, or Lightning connectors
- Bluetooth speakers are charged via 5V USB-Micro connectors

Wireless Charging

There are two commonly-used methods of wireless charging — inductive charging and resonant charging.

With inductive charging, energy is transferred from the power supply to the device via an electromagnetic field. Energy from the power supply is used to create an electromagnetic field that transfers energy wirelessly to a magnetic receiver connected to the battery of the device being charged. The magnetic receiver on the device creates a direct (DC) current that charges the battery inside the device. Wireless charging can travel through most materials except for metals, so it is ideal for devices enclosed in plastics. Examples of wireless charging include electric toothbrushes, Amazon Tap, or QI chargers for smartphones.

Resonant charging is similar in most ways to inductive wireless charging, except that it has the advantage of being able to charge devices over greater distances than inductive charging. Because power can be transferred without direct physical contact, it also enables charging multiple devices at the same time.

Although both wireless inductive and resonant charging have distinct advantages because they do not require a cable or physical interface for energy transfer, they both have a lower charging speed than wired charging.

Intermittent Power

Intermittent power generation is any charging that occurs from a source which is not continuously available. The availability of intermittent power is often variable (e.g. solar on a cloudy day). In most cases, intermittent power that is generated must be stored, typically in a battery, for later use.

Solar Panels

One of the most common examples of intermittent power generation is with the use of solar panels. Solar panels work by taking sunlight (indoor lighting is typically not sufficient) and converting it into electrical energy that can then be stored in a battery or bank of capacitors. Solar panels work best when they are clean and directly oriented towards the sun. The size of the solar panel will determine the speed at which it is able to recharge an external battery or bank of capacitors that can be used to store the power that is generated. Solar panels are typically rated by their maximum power output, which is measured in watts (wattage = voltage * current).

Particle Example: Opti, powered by solar power and our Electron enables storm water management by anticipating the levels of water from weather reports, and emptying out the reservoir in time for the storm drains to be effective.

Energy Harvesting

Another example of intermittent power generation is energy harvesting. Energy harvesting refers to mechanisms that increase the power efficiency of a device by recollecting small amounts of energy that would otherwise have been lost as light, heat, sound, or vibration.

Because energy harvesting sources generally have low energy potential and low conversion efficiency, energy harvesting is most effective in applications where there are large amounts of waste energy that an IoT device can tap into. For example, a device operating in a chemical plant or connected to a gas turbine has a large amount of waste heat that it can convert to usable power.

Non-regenerative systems

If your device has a battery that is not rechargeable (i.e. alkaline batteries) or does not take advantage of intermittent charging, it is important to alert the user in a timely manner so that the device's battery can be recharged or replaced. When designing an Internet-connected product, these alerts can be delivered automatically to the device's primary user interface (mobile or web app) that is used to configure and control the device.

Conclusion

When designing an IoT product, it is important throughout the design process to consider how a device's behavior will effect its energy budget. Energy consumption, battery capacity, and power generation are key components of the energy budget.

It is possible to extend the life of a product by reducing energy consumption. Energy usage by the microcontroller and radios can be reduced through several strategies including use of sleep modes and minimizing data transmission. Smart selection of components can also drastically reduce energy use. Because low power applications are so common for IoT and connected devices, Particle hardware and software supports these design strategies and provides flexible and intuitive engineering tools for getting the most customer and business value from a single charge.

Once the rate of energy use for an optimized product has been determined, it is then possible to select a battery for the product and choose a power replenishment strategy.

For more information about power management, visit: docs.particle.io and search for "Sleep"