

Power management

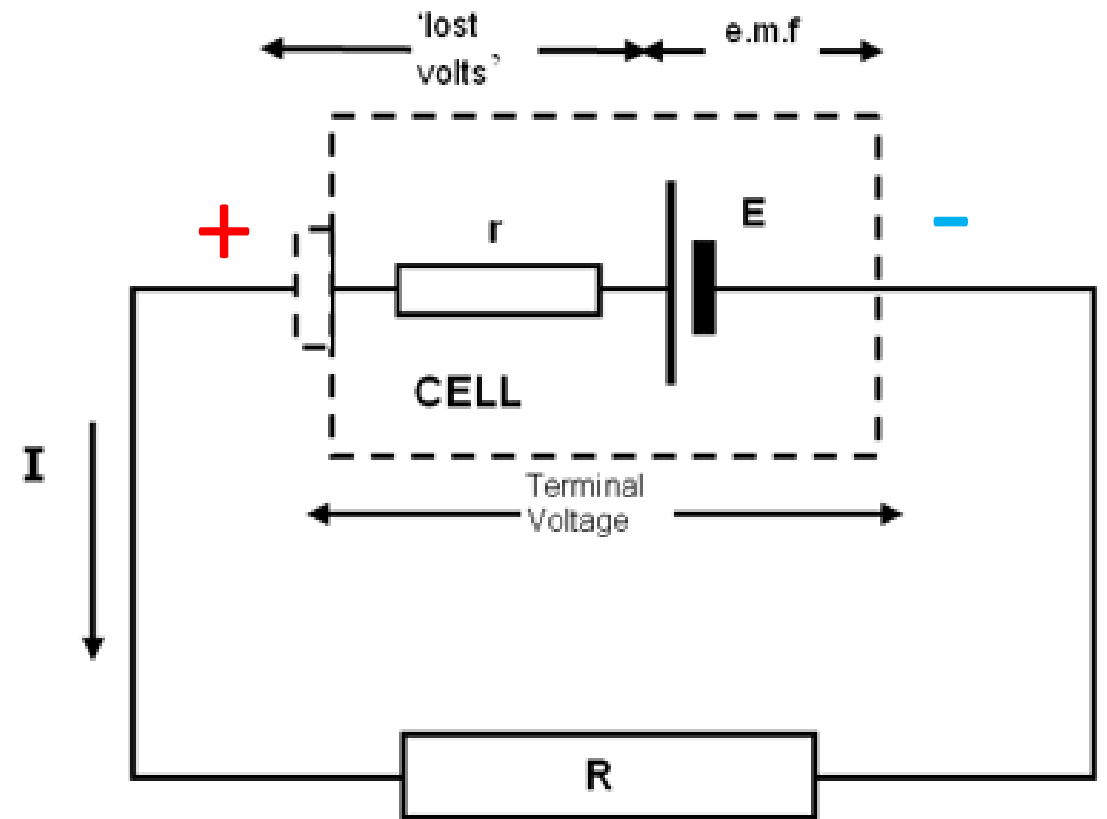
Agenda

- Power supply
- Batteries
- ESP32 sleep modes

Power supply

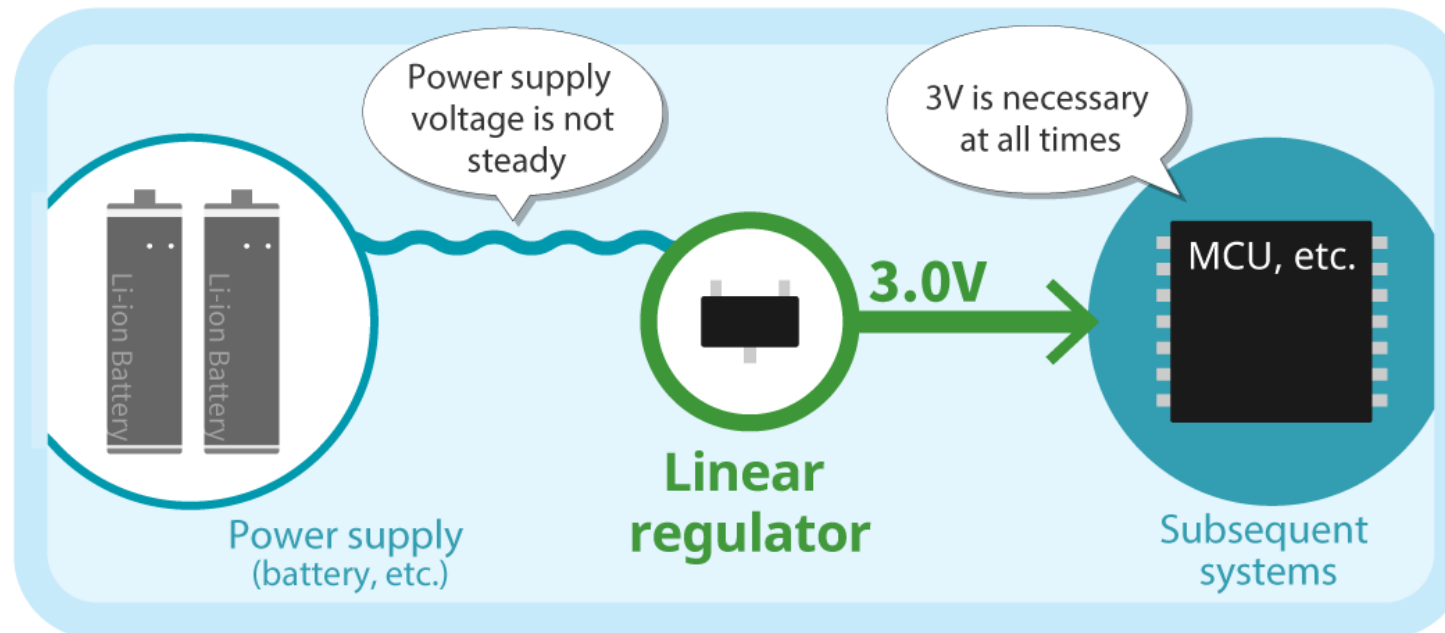
Power supply

- Sources
 - Grid, Battery, Solar
 - Generator, Peltier, RF harvesting
- Main characteristics
 - Rated voltage and current
 - Polarity
- Undesired effects
 - Noise / Fluctuations
 - Internal resistance



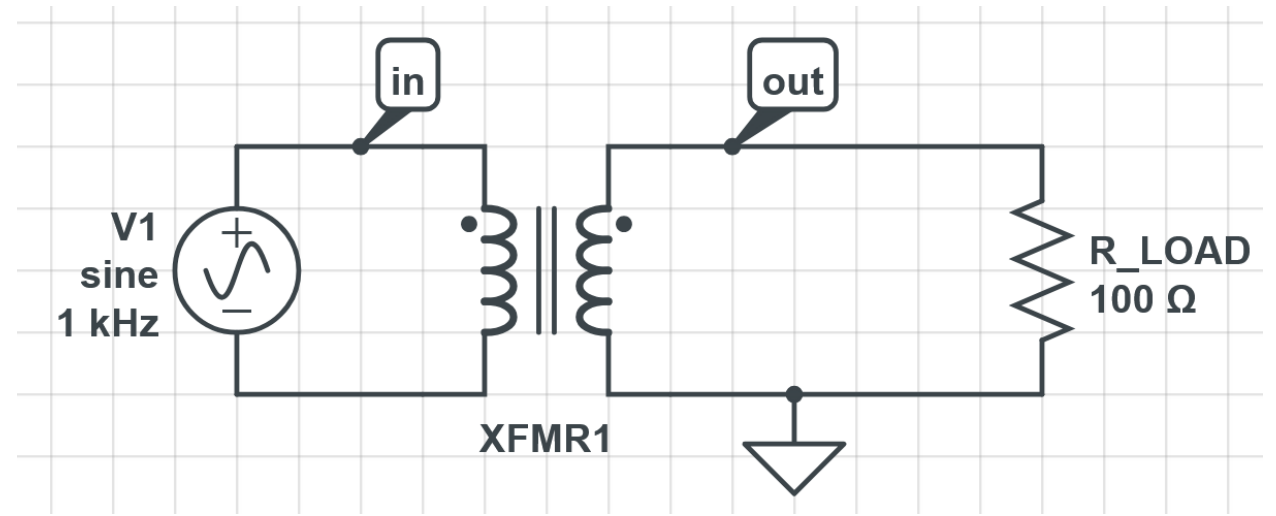
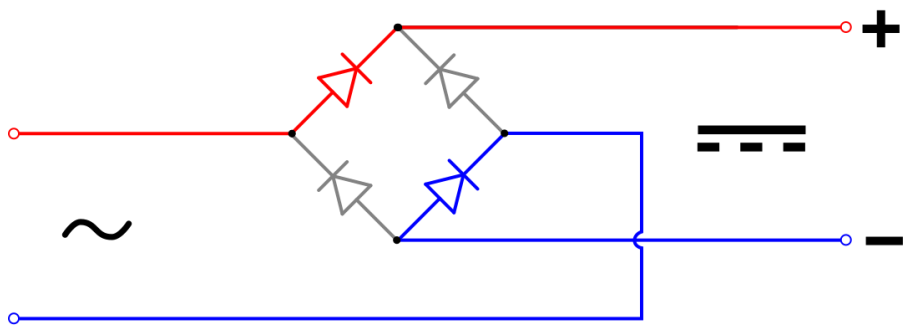
LDO: Low dropout (aka linear) regulator

- Stabilize **small** voltage fluctuations
- Low power (power the MCU, sensors ...)
- Low noise, but low efficiency (voltage drop = power loss)



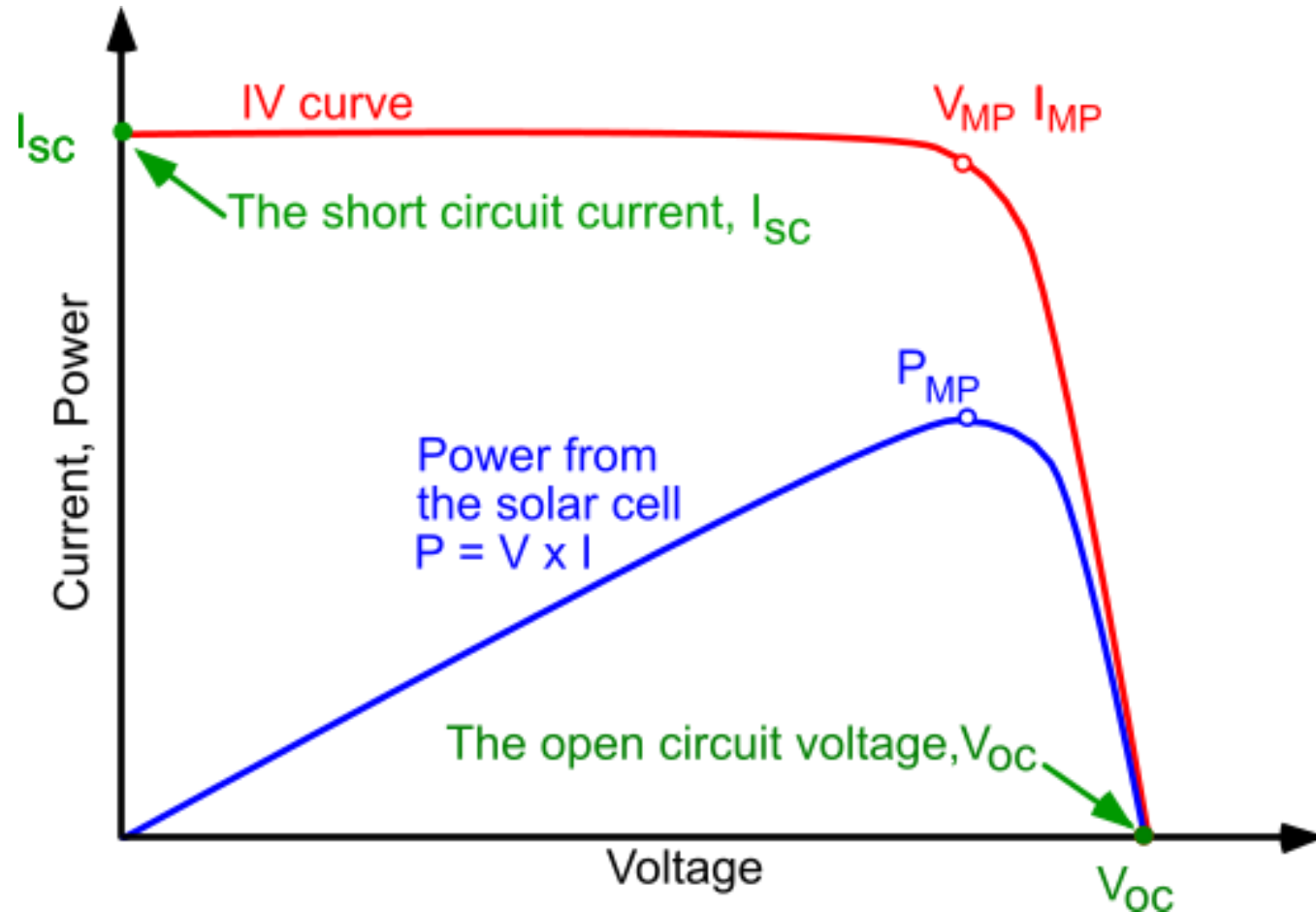
Transformers

- Purpose: step up/down the voltage
- Type: AC (sine) vs Switched
- Rectifiers (Gretz bridge)



Solar (PV) panels

- Open circuit voltage (V_{oc})
- I/V curve (at given irradiance)
- Maximum Power Point
- Heat management
- Inverters



Batteries

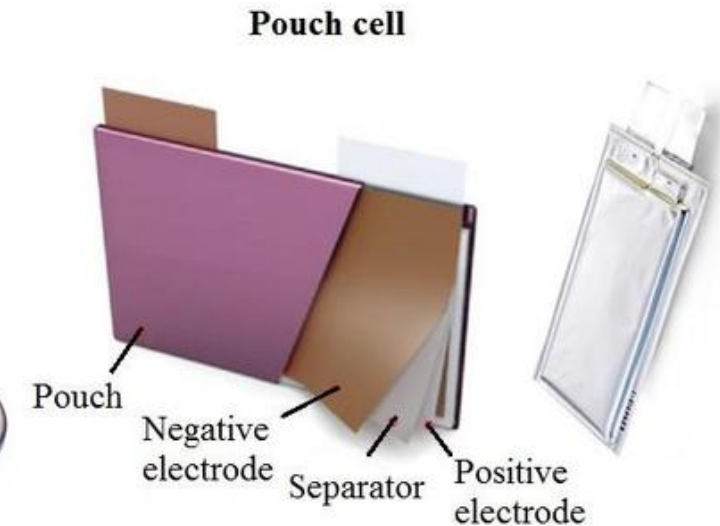
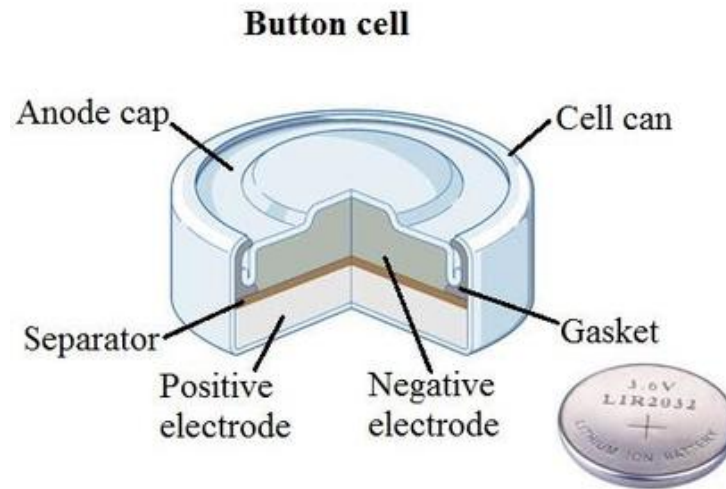
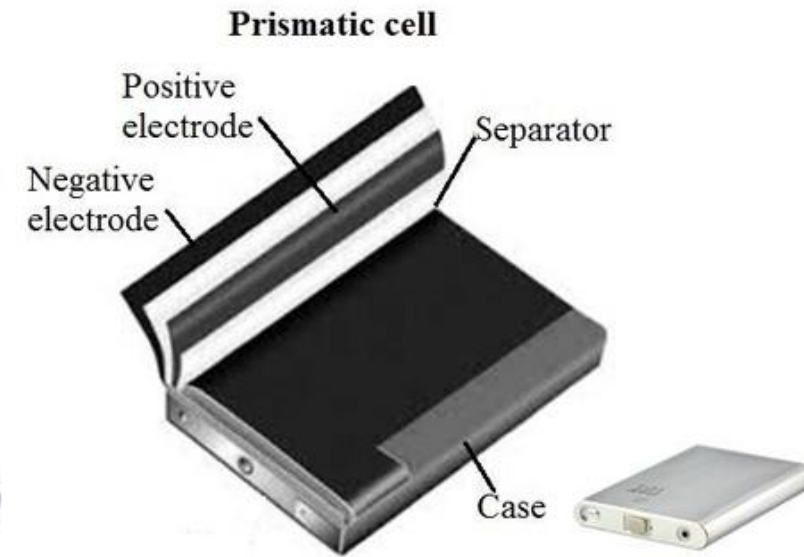
Common batteries

Type	Voltage (V)	Capacity (mAh)	Resistance	Self Discharge
Coin Cell 2032	3.00	200	High	Low
AAA/AA/C/D (Alkaline)	1.2 – 1.5	1000/2000/5000/10000	Medium	Low
AAA/AA/C/D (NiMh)	1.2 – 1.3	1000/2000/5000/10000	Medium	Med/Low
AAA/AA/C/D (Ni-Zn)	1.3 – 1.6	1000/2000/5000/10000	Medium	Med/Low
LiPo - 18650	3.3 – 4.2 (3.7)	2500-3500	Low	Low
LiPo – 21700	3.3 – 4.2 (3.7)	4000 – 5000	Low	Low
LiPo – 14500	3.3 – 4.2 (3.7)	500 – 1000	Low	Low



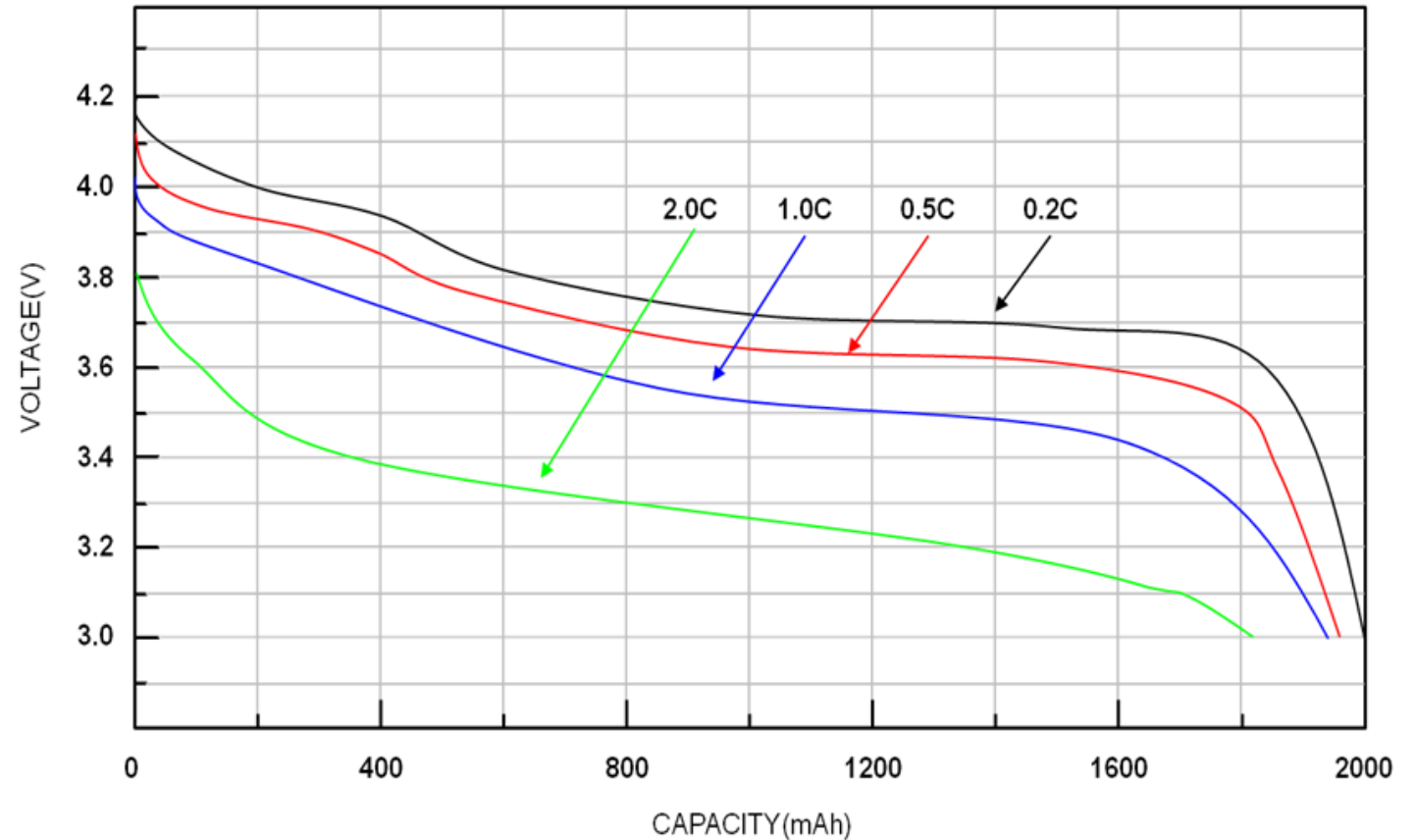
LiPo form factors

- Cylinder
 - higher density
 - cheaper
 - more robust mechanically
 - safety vent
- Pouch/prism
 - cools better
 - flat devices (phone, laptop)
 - swelling is a problem



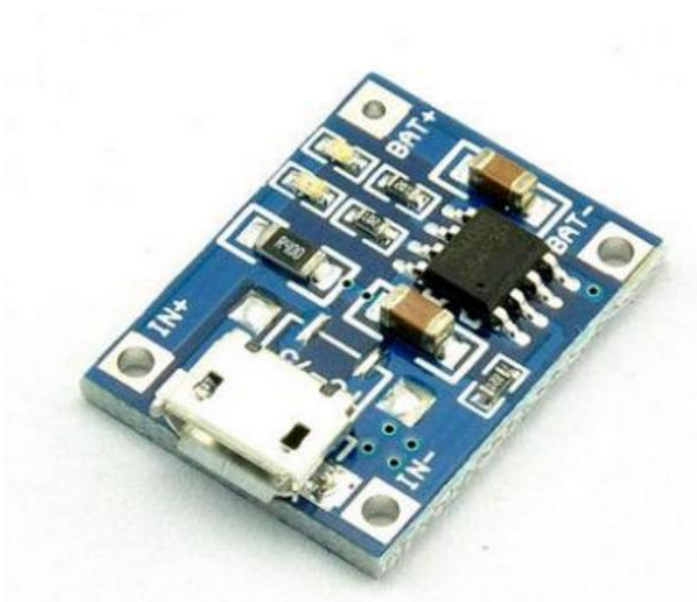
LiPo Discharge Curve

- Internal Resistance increases with decreased capacity
- 1 ohm @ 300 ma = 0.3v voltage Drop
- Discharge below 3V may damage the cell

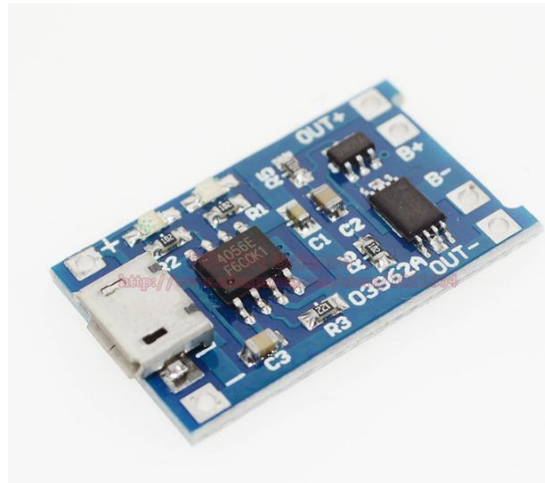


LiPo chargers

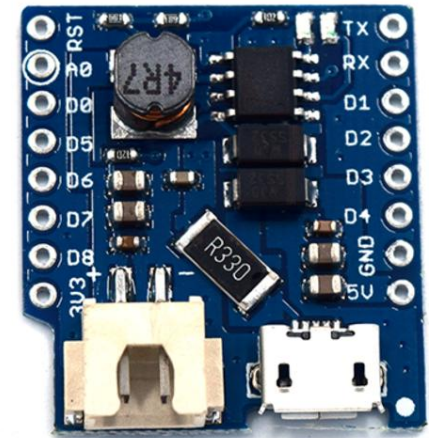
- Only Charge



- + battery protection



- Wemos Battery Shield
 - + battery protection
 - + 5v Boost



ESP32 sleep modes

ESP32 sleep modes (datasheet)

Power mode	Description			Power Consumption
Active (RF working)	Wi-Fi Tx packet			~100–240 mA but bursts to 790 mA
	Wi-Fi/BT Tx packet			
	Wi-Fi/BT Rx and listening			
Modem-sleep	The CPU is powered up.	240 MHz [*]	Dual-core chip(s)	30 mA ~ 68 mA
			Single-core chip(s)	N/A
		160 MHz [*]	Dual-core chip(s)	27 mA ~ 44 mA
			Single-core chip(s)	27 mA ~ 34 mA
		Normal speed: 80 MHz	Dual-core chip(s)	20 mA ~ 31 mA
			Single-core chip(s)	20 mA ~ 25 mA
Light-sleep	-			0.8 mA
Deep-sleep	The ULP coprocessor is powered up.			150 μA
	ULP sensor-monitored pattern			100 μA @1% duty
	RTC timer + RTC memory			10 μA
Hibernation	RTC timer only			5 μA
Power off	CHIP_PU is set to low level, the chip is powered down.			1 μA

Modem sleep mode (automatic)

- Router (AP) broadcasts DTIM beacon regularly (100-1000 ms)
- ESP32 (STA) tunes to the beacon timing and wakes up Radio on time
- Power consumption: ~20mA (modem sleep) to ~240mA (active)
- CPU clock is adjusted automatically

Light sleep mode

- CPU, RAM and peripherals are **clock-gated** (paused), voltage reduced
- RTC and ULP remain active
- Power consumption: **~1mA**
- No data loss, program resumes upon wake up

Deep sleep mode

- CPU, RAM and peripherals are **powered down** (data lost)
- RTC and ULP remain active (RTC RAM preserved)
 - Example: `RTC_DATA_ATTR int myVar = 0;`
- Power consumption: **~10-150uA**
- Wake up resets the CPU
 - “deep sleep wake stub” callback can alter the boot process

Hibernation mode

- CPU, RAM and peripherals are powered down (data lost)
- ULP and 8MHz Oscillator (clock) powered down
- Only RTC timer and RTC GPIOs remain active (**RTC RAM lost**)
- Power consumption: **~2.5uA**
- Wake up resets the CPU

Wake up sources

- **Timer:** `esp_sleep_enable_timer_wakeup (microseconds);`
- **Touch (interrupt):** `esp_sleep_enable_touchpad_wakeup();`
- **External wake up (interrupt)**
 - ext0: a single, preconfigured GPIO pin interrupt
 - ext1: any RTC pin, pull up/down not available, RTC memory not required

```
esp_sleep_enable_ext0_wakeup(GPIO_PIN, LOGIC_LEVEL)
```

```
esp_sleep_enable_ext1_wakeup(BUTTON_PIN_MASK, LOGIC_LEVEL)
```

ULP coprocessor

- Uses (SLOW) RTC memory
 - A big enough chunk must be reserved for ULP code and data
 - Program(s) embedded into main program as a blob
 - And loaded to RTC memory with : `ulp_riscv_load_binary` / `ulp_riscv_run`
- Code is triggered periodically by a timer (136KHz RTC_SLOW_CLK)
- Can share variables with the main program
 - Synchronization: `ulp_riscv_lock_acquire/release`
- Can wake up main CPU
- Can read sensors (I2C, bit banging) / GPIOs / ADC
- Programmed in C or assembly (RISCV)