

Mote Hardware and Energy Consumption

Communication

The most important parameter for a mote radio is the effective energy per delivered bit, but there are a lot of other related and important parameters. We'll see in later chapters how range is related to transmit power and receiver sensitivity, and how network lifetime is related to range and radio current consumption. For now we'll just look at the basics.

Transmit power, Transmit Current

Typical transmit power (the power actually radiated out the antenna as RF energy) is in the 1 to 10mW range. The radio that generates this transmit power can be thought of as having two parts: the modulator, which converts bits into the appropriate time varying (radio frequency) voltage, and the power amplifier that boosts that signal up and delivers it to the antenna.

In low power radios, the modulator often burns more power than the power amplifier. With the power amplifier off, essentially no power goes out the antenna, so there's a minimum "overhead" of current from the power supply just to generate the appropriate voltages. It is difficult to design a power amplifier that is efficient over a wide range of output powers, and power amplifiers are generally designed to be most efficient near their maximum power output capability. The efficiency of the PA is typically between 10% and 80%, and the lower end of that range is most common for low power chips.

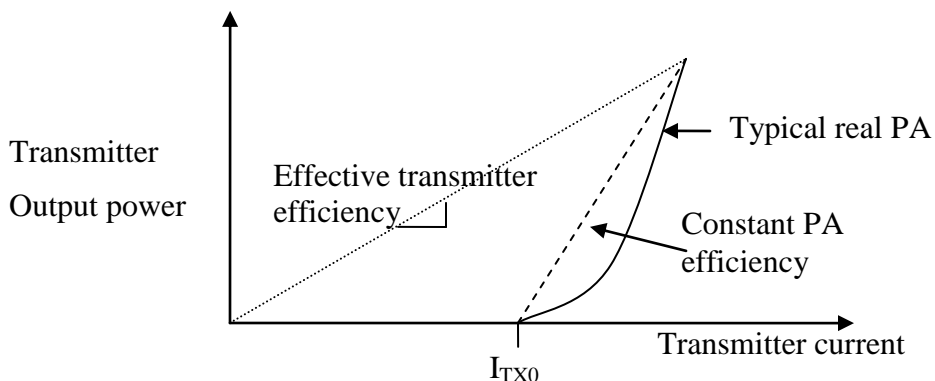


Figure 1 Transmitter output power versus input current.

The result is that a 10x reduction in the output power of the radio is rarely coupled to a corresponding reduction in radio current. Although most sensor network radios do have some type of transmit power control, often over two orders of magnitude or more, the difference in radio current is rarely greater than 2x.

Receive sensitivity and current

Radios can only receive information if the received signal is strong enough. The minimum detectable signal level for a radio is called the sensitivity. Typical numbers for mote radios are a fraction of a picoWatt. The ratio of the transmit power to the receive

sensitivity is called the link margin. For a transmitter putting out a few milliWatts, and a receiver with a sensitivity of a few tenths of a picoWatt, the link margin is around 10 billion! In principle, one transmitter could talk to ten billion receivers, if the transmit power were evenly divided. In practice, we'll see that sometimes it can be challenging to receive a message from a mote that's only a few meters away. The mechanism by which the RF energy leaves the antenna, propagates through space, and arrives at the receiver antenna is the topic of the next chapter.

Bit rate

Motes use digital radios that send information in packets, which are collections of bits. The bit rate varies from thousands of bits per second to millions of bits per second (kbps to Mbps). The transmit current divided by the bit rate yields the amount of charge pulled out of the supply per bit transmitted, Q_{TXbit} , since $[mA/kbps] = [\mu As/bit] = [\mu C/bit]$. Similarly, receive current divided by bit rate yields charge consumed per bit received. Multiplying by the supply voltage yields the energy per transmit E_{TXbit} and receive (RX).

The IEEE 802.15.4 standard specifies a 250kbps data rate in the 2.4 GHz band. This is the most common radio standard for wireless sensor networks. The original 15.4 chips burned about 20mA in transmit and receive, for a Q_{TXbit} and Q_{RXbit} of $20mA/250kbps = 80nC/bit$. Operating at 3V, these radios burned 240nJ/bit.

802.11g chips from Atheros run at up to 108Mbps, and burn perhaps 200mA, for a Q_{TXbit} of only 2nC/bit. Does this mean that we should use .11g chips for low power networks? The answer is not simple. As we will see later, the medium access protocol will be an important role in the optimum choice of radio, but in addition there is the reality that radios do not turn on instantly.

Radio startup

Before the first bit can be sent or received on the radio, a long sequence of events must typically take place. From a deep sleep state, this entails turning on a voltage regulator, waiting for a crystal oscillator to stabilize, and waiting for the radio oscillator to settle (tune) to the proper frequency, among other things.

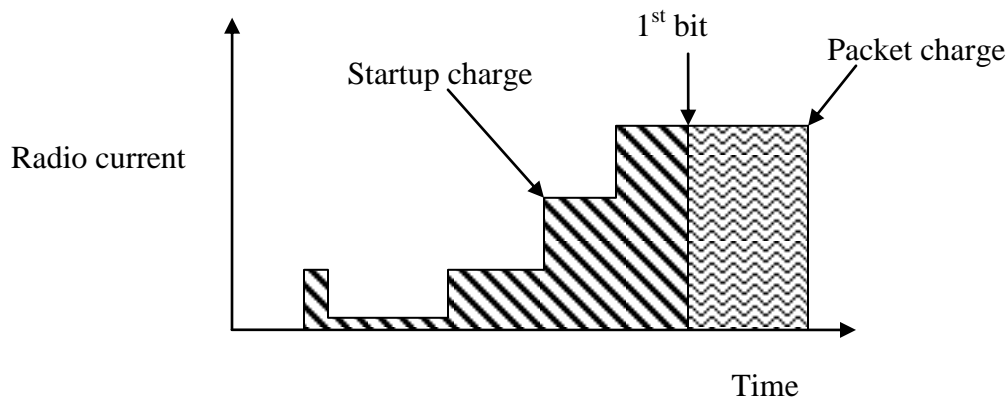


Figure 2 Radio current during startup and TX.

In a plot of radio current from deep sleep, through startup, to transmission of a packet, the area under the curve before the first bit is transmitted is the startup charge, Q_{TX0} , and the total charge for the packet is the startup plus the number of bits in the packet times the charge per bit. $Q_{TX} = Q_{TX0} + n_b * Q_{TXbit}$

Computation

Performance

Low power microprocessors typically operate with 8, 16, or 32 bit quantities of data. Usually the instruction width is the same as the data width, although there is now a family of 32 bit processors from ARM that use a 16 bit instruction set.

In general, the wider the datapath and instruction, the more you can do in a single instruction, and a single clock cycle. So a 20MHz 8 bit processor will be a lot slower than a 20MHz 32 bit processor (sometimes more than 10x), and the code size for the 8 bit processor will be larger than for the 32 bit processor (maybe tens of percent).

The energy per cycle is the processor current ($\text{mA} = \text{mC/s}$) times the voltage, divided by the clock rate (cycles/s). [$\text{mC/s} * V / \text{cycles/s} = \text{mCV/cycle} = \text{mJ/cycle}$]. It's usually fairly constant over at least an order of magnitude of clock frequencies, meaning that the current is proportional to clock speed. At low frequencies, the current flattens out at some minimum non-zero value, so the energy per cycle goes up. Typical numbers for well-designed low-power processors are below 1nJ/cycle.

For most modern low-power processors there is roughly one cycle per instruction, so we get 1nJ/instruction.

Since there is such a wide range of performance per cycle, it's much more interesting to figure out what the energy per operation or function is. For dedicated hardware blocks such as encryption, this is number is generally fixed and available. For less-well-defined operations like "prepare a data packet for transmission", the results will depend on the quality of the code and compiler as well as the microprocessor.

Memory

Programs are typically stored in flash memory or ROM. Both of these are non-volatile, meaning that they don't go away when the power is turned off. ROM and flash are both very high density, (roughly 1Mb/mm^2 in $0.18\mu\text{m}$ CMOS).

Variable data is stored in SRAM, which is about ten times lower density than flash or ROM, so typically there's a lot less of it on a microprocessor. SRAM is volatile, so when the rest of the chip goes to sleep, you still have to keep your SRAM powered if you want to retain any of the information in it. Fortunately, it doesn't burn much power when it's just sitting there retaining information, since it's just the leakage through all of the transistors. A few μA is typical, although the leakage current is exponential in temperature, and can get into many tens of μA if the SRAM isn't specifically designed to be low leakage.

Some hardware platforms use external flash memory, typically with a serial interface. This costs a lot more to access than on-chip memory, but still typically a couple of orders of magnitude less than sending the data over the radio.

Oscillators, timers, interrupts, and sleep states

Most motes will keep a real time clock (RTC) running at all times. This is almost always based off of a 32,768 Hz crystal oscillator just like those found in wristwatches, and virtually all electronic devices. Driving this oscillator, and running a digital counter off of it to keep track of time, typically burns about 1uA. When a timer is set in software, somewhere in hardware the time will be converted to ticks of the 32kHz oscillator, added to the current time, and put into a special Timer register (TR). Another piece of hardware continuously checks to see if the current time is equal to the value in the TR. When the two are equal, an interrupt is generated that causes the processor to execute the appropriate code.

Most mote processors spend most of their time sleeping. The software wakes up periodically due to timer interrupts to, e.g., sample sensors and send or receive packets, and when it's done it goes back to sleep. Using a processor that efficiently moves between sleeping and waking states is important for low power operation.

Sensing

There are thousands of different kinds of sensors, each with its own interface specifications. Even a single company, selling a single type of sensor, may have dozens of different versions of that sensor with different performance, interfaces, packaging, temperature tolerance, etc.

There is no such thing as a general sensor interface. Increasingly, sensors have integrated electronics, so they may present either a digital interface or a low impedance voltage output, both of which are relatively easy to interface to a microprocessor.

Energy per sample

Taking a sample is a multi-step process, with two primary components: sensor excitation, and sample acquisition. Most sensors require some type of electrical power input before they can operate, and there is a minimum time for them to stabilize before the output of the sensor is valid. The integral of the power over the startup time is the minimum energy per measurement.

The output of the sensor may require amplification or filtering, both of which will have their own power and settling time requirements. Fortunately, for most sensors, either the power or the settling time are small, and the energy per measurement is low. For example, a pressure sensor or accelerometer with a piezo-resistive bridge sensing element may burn tens of milliWatts when it's on, but it can settle in a fraction of a microsecond, so the total energy per sample can be tens of nJ. An electrochemical gas sensor may take seconds or minutes to equilibrate, but burns no power at all, since it is the presence of the measurand that causes a chemical reaction, which generates the electrical energy that is measured. GPS is one of the high-energy exceptions: it takes tens of milliWatts to run, and requires at least seconds to acquire a reading, leading to tens of mJ of energy per

sample.

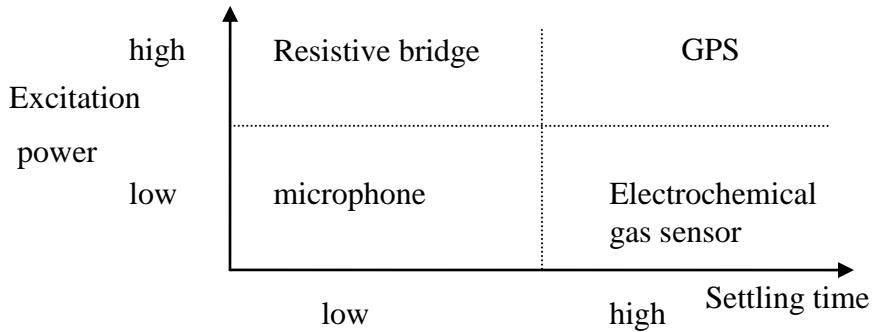


Figure 3 Relative excitation power and settling time.

In parallel with sensor excitation, the ADC must be initialized. This process will typically involve turning on an analog voltage reference, which will take some power and initialization time. The voltage reference and/or voltage regulator associated with this process may have substantial capacitance in their designs. If they weren't designed to be low energy per sample, then the amount of charge required from the battery for each initialization can be substantial. For example, a 10uF capacitor charged to 3V requires 30uC from the battery. If this charge disappears after each sample, then turning the ADC on once per second will cost 30uA. This is still probably much lower than the cost of just leaving the ADC on between samples.

MEMS sensors

It is very common now for MEMS sensors to be integrated with their own analog to digital circuitry and compensation logic. They present their sensor value in digital format over a serial port. This makes interfacing to the sensor easy, but sets a lower bound on the sample energy which may not have been a design consideration.

Battery Power

While there are many sources of power and energy, in this section we'll just cover batteries, and leave energy scavenging for another chapter.

Chemistry

The battery anode and cathode chemistry determine its fresh (pre-discharge) open circuit voltage as well as the temperature range for normal operation.

Chemistry	V _{OC} , 25C	Temp range [C]		
Alkaline	1.6			
CR - lithium MnO ₂	3	-30 ... +60		

BR - lithium (CF) _x	3			
Li SOCl ₂	3.6	-55 ... 125		

Lithium batteries have a flat discharge profile, meaning that their voltage remains constant over most of their useful life. Alkaline batteries have a linear decrease in voltage as their capacity is drained. Lithium batteries generally have longer shelf life due to lower internal leakage.

Lithium thionyl chloride batteries have the highest energy density, largest temperature range, and lowest internal leakage of any of the commonly used batteries. Some companies have had lithium thionyl chloride batteries in service for over two decades.

Consumers pay much higher prices for lithium batteries than for alkaline. In high volume, CR and BR chemistries are still more expensive than alkaline, but the difference is not quite as dramatic. Lithium thionyl chloride are the most expensive of all, befitting their highest performance.

Package

For a given chemistry, the package determines the total charge stored in the battery, and has a strong influence on the maximum current. The most familiar sizes of alkaline “cylindrical cell” batteries are AAA, AA, C. Many lithium chemistries come in these sizes as well. In addition, lithium batteries come in “coin cell” sizes as well, the most popular of which is the 2032, and the largest of which is the 2477. The first two digits are the diameter of the cell in millimeters, and the second two digits are its thickness in tens of millimeters. The smallest coin cells are called button cells, and use different naming conventions. The smallest of these are for hearing aids.

Lifetime

In addition to the current that you pull out of the battery, it will also self-discharge. This limits the lifetime for low current levels. At high currents, the internal resistance of the battery, among other things, reduces the amount of charge available to the application.

The useful capacity of a battery is a strong function of temperature, average current, and current profile.

Problems

1. Look at the datasheet for the Panasonic CR2477. What measurements do you need to make to tell me how much charge is left in the battery? How accurate do you think your estimate is?
2. If you have a 250kbps radio with a TX current of 25mA, and you want to sample a sensor once per second and send the 8 bit result over the radio, calculate the average current under the following assumptions

- a. There is no energy consumed to acquire the sample. There is no startup time for the radio – you just wake up and transmit.
 - b. As in a, except the radio burns an average of 10mA during a 1ms startup period.
 - c. As in b, except you decide to collect samples for two minutes at a time, and send all 120 samples at the end of each two minute period.
 - d. As in c, except
 - i. The 32kHz oscillator and related timers require 1uA of current.
 - ii. The sensor requires 1mA excitation for 500us to stabilize before the sensor reading.
 - iii. The analog regulator for the ADC burns 100uA and requires 1ms to stabilize, and has a 10uF capacitor which must be charged to 1.8V before the sample can be taken.
3. Calculate the lifetime of the various versions of the system in the problem above assuming the following batteries and conditions. Pick a particular battery and use its online datasheet.
- a. a CR2032 battery operating at room temperature
 - b. a AA alkaline battery operating at room temperature
 - c. a AA alkaline battery operating at 70C