ambiq micro
Giving batteries a better life

White Paper

Reducing Power in Wearable Electronic Devices

ambiq micro
Giving batteries a better life

Reducing Power in Wearable Electronic Devices

The electronic revolution that started with the smartphone is experiencing a new wave of innovation: wearable electronics. Enabled by many of the same technologies that power a smartphone, they are nonetheless distinct, with their own unique use cases and design constraints.

When asked about their biggest challenges, wearable electronics designers tend to list the following three items at the top of their lists:

- **Battery life** – a weakness in all current wearables.
- **Battery size** – this ends up driving the form factor more than any other single component.
- **Additional functionality** – new and useful functions which cannot currently be easily accommodated due to power limitations.

Power consumption lies at the root of all three.

The smartphone originated as a telephone, and as such, users have become accustomed to charging the phone daily – or even multiple times a day – so that they can make phone calls. Battery life is something of a selling feature, although visible more in customer reviews than in sales materials.

Wearable devices have much stricter power requirements than a smartphone. Exactly how strict depends on how it's used. If it's meant to be taken off at night, then a nightly charge might be tolerated. But great effort is going into the physical design of wearables in order to make them more comfortable and attractive – even to fade into the background. A wearable that the consumer can forget about is also one where charging must be a rare event. Such a device must sip power much more gently than a smartphone.

Wearables will also increasingly require a feature that may also be expected of smartphones in the future: "always on." Just as we expect a watch to keep running whether or not we're looking at the time, we will also expect our wearable electronics to keep doing what they do whether we're paying attention to them or not. All without requiring a frequent battery charge.

There's one other major difference between a smartphone and a wearable device: the application processor (AP). The phone has one, and power-savings efforts frequently focus on offloading tasks from the power-hungry AP to some more miserly processor so that the AP can be put to sleep. But

most wearable electronics have no AP. That's helpful in that it removes a big consumer of energy, but it also means that phone-oriented power-saving strategies may not apply to a wearable device.

ambiq micro
Giving batteries a better life

In the abstract, all modern wearables have the same high-level architecture: their role is to measure something, perform some computation on the measurements, and then send some result – typically wirelessly – to another device – typically a smartphone.

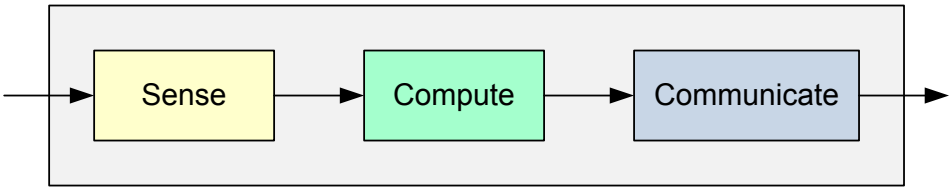

Figure 1: Abstract wearable device architecture

In the future, this could also include an actuation step – perhaps medication might be released, for example. But the current generation of wearables doesn't attempt that, so it won't be included in the discussion that follows.

The remainder of this document will walk through this architecture. First, each block will be examined for factors affecting power consumption, and then system-level optimizations will be considered – since the boundaries aren't as solid as they look.

Important note: the following sections contain actual current numbers from sample devices. The intent is solely to give some real examples with real numbers. The intent is NOT to do competitive comparisons.

- The devices chosen should not be assumed best in class. Assume that the samples were chosen at random amongst devices appropriate for the application.
- The numbers shown below are pulled directly from datasheets. If you're trying to compare devices for selection purposes, you should acquire the full datasheets.
- Just because one device might draw less current than another shouldn't lead to a conclusion about which devices is "better." There are numerous other important features and caveats, not least of which is that testing conditions are likely to vary by vendor. These tables should not be considered a shortcut to device selection.
- Microcontroller current-measuring conditions vary so widely by vendor that some work has been done to normalize the numbers to the same set of conditions.
- Device availability and numbers are accurate as of when this was written. Consult current datasheets for any changes.

Sensors

Most wearables today limit themselves to one sensor: an accelerometer. But there's a desire to incorporate other sensors in the future, so the following descriptions cover those sensors most in demand for addition to wearables.

Typical consumer-grade sensors employ microelectromechanical system (MEMS) technology to achieve the size and cost levels required for use in a wearable device. All MEMS sensors will include an application-specific IC (ASIC) for cleaning up and digitizing the raw analog sensor readings and for sensor control. For most MEMS sensors, the MEMS element does not require power, but the ASIC will. There is, therefore, no zero-power sensor.

Some sensors will provide analog outputs; some will provide digital outputs. The latter will typically consume more power since there is more circuitry (although it's not always possible to find pairs of sensors that are identical except for output type, so exact comparisons can be tough).

Sensors with digital outputs are typically accessed by either the I^2C bus or the SPI bus. Some sensors have only one such port; some have both. The bus type has no appreciable effect on power consumption.

The outputs of different sensors can often be combined to create higher-level data. For example, magnetometers and gyroscopes can be combined to yield more reliable orientation information; accelerometer data and orientation can be combined to create position information in quaternion form. Such combining algorithms are generically referred to as "sensor fusion"; this will be discussed in more detail below.

Power is an important feature for many sensors, so you will typically find datasheet sections dedicated to managing power. In particular, most sensors intended for power-sensitive applications will have one or more power-down modes. The following considerations apply regardless of the sensor being evaluated:

- Different manufacturers will create different power modes. This is often a point of competition, so you should expect manufacturers to differentiate with their power modes. Review them carefully, since identically-named modes may operate very differently from one vendor to another.
- Power modes may affect numerous parameters, including sensitivity, precision, accuracy, signal-to-noise ratio, access to internal registers (reading and/or writing), register state preservation, and wake-up or state-transition times. Modes must be selected in consideration of the limitations they impose.
- Power will depend on the sensor sampling frequency and the rate at which the sensor data is read.
- n order to reduce the frequency of output read operations, many sensors feature FIFOs that can accumulate multiple sensor samples for less frequent batch reading.
- Datasheets often give only typical currents. When comparing devices, it's important to note whether numbers reflect typical or maximum currents.

The quality of the information available in sensor datasheets varies widely, and it can frankly be frustrating to try to compare individual devices. But to truly minimize power, it is worth the time to understand devices in detail both to select the device and then to optimize its use.

Given a particular vendor, it may not be obvious which sensors or families have the lowest power. Power is often not shown on selection tables, which means that you may have to check each product individually.

As an alternative, some distributors have excellent product selection tables with filtering capabilities that may allow you to find a device in your power range more quickly.

Accelerometers

Accelerometers are the most fundamental of the motion sensors. Having been miniaturized through MEMS technology, they consume little power and are inexpensive for consumer-grade accuracy.

Because they consume little power, accelerometers can be used to wake other sleeping motion sensors when they're needed. This typically involves putting the other sensors into some sleep mode and then putting the accelerometer into a low-power mode where it can detect basic motion with low resolution.

When motion is detected, wake-up must happen quickly enough to minimize the loss of initial motion detail while the sensors are waking up.

Most consumer-grade accelerometers integrate three axes of measurement into a single sensor package. These save space, power, and assembly cost, as compared to using discrete single-axis sensors. Other sensor combination will be discussed below.

Example devices:

Device	Mode	Current	Notes
Analog Devices ADXL362 (3-axis)	Measurement	1-5 µA (typ.)	Save power by reading only the 8 most-significant bits
	Wake-up	270 nA (typ.)	6 samples/sec; for detecting basic motion
	Standby	10 nA (typ.)	Sampling disabled
STMicroelectronics LIS3DH (3-axis)	Normal	11 µA (typ.)	Full resolution
	Low-power	6 µA (typ.)	Reduced resolution
	Power-down	500 nA (typ.)	(Full characteristics not included in datasheet)

ambiq micro
Giving batteries a better life

Power-saving tips:

- Select an accelerometer designed for low-power use.
- Leverage power modes.
- Use FIFOs for batch reading.
- Use the accelerometer to wake other sensors.
- Consider combos (see below) if other sensors are needed in addition to accelerometer

Gyroscopes

Accelerometers measure linear acceleration; gyroscopes measure rotational acceleration. Together, they provide data for all six degrees of freedom for motion. The challenge with gyroscopes, however, is that their MEMS elements must be actively powered to detect angular changes. This makes them high consumers of power – and they are often resisted as a result.

You may actually be able to get away without a gyroscope. If you assume, for instance, that motion always applies to a pedestrian moving forward, then you also assume that the pedestrian is always facing in the direction of travel. That simplifying assumption means that you can rely solely on the accelerometer to detect turns without independently sensing changes in orientation. If the path turned left, then you've assumed that the pedestrian has also rotated to face left. If, on the other hand, you wish to detect orientation independently of travel direction, then a gyroscope is indicated.

Gyroscopes tend to have more elaborate power modes. A common arrangement is to put the gyroscope to sleep when possible, and have the accelerometer remain in a low-power wakeful state. When the accelerometer detects motion, it wakes the gyroscope.

Gyroscopes can measure very quick, subtle orientation changes, but they also suffer from drift. There are a few configurations where drift can be zeroed out often – like in a shoe, where, with each footfall, you know that the shoe is exactly at rest. Elsewhere, however, the gyroscope is in constant motion, and the drift error accumulates in a matter of seconds (especially for low-cost consumer-grade gyroscopes). Gyroscope manufacturers work to keep their drift numbers competitive, but a magnetometer can act as a cross-check through sensor fusion, correcting the drift. Magnetometers are discussed in the next section.

As with accelerometers, consumer-grade gyroscopes integrate three axes of measurement into a single sensor package. Other sensor combination will be discussed below.

ambiq micro
Giving batteries a better life

Example devices:

Device	Mode	Current	Notes
Bosch BMG160 (3-axis)	Normal	5 mA (typ.)	
	Fast power-up	2.5 mA (typ.)	No sampling; sensing analog off; digital/drive on
	Suspend	25 µA (typ.)	All analog off; write restrictions
	Deep suspend	< 5 µA (typ.)	Register values lost
	Advanced	Varies	Cycles between normal and fast-power-up modes; power depends on adjustable timing and duty cycles.
InvenSense ITG-3701 (3-axis)	Normal	3.3 mA (typ.)	
	Power-down	8 µA (typ.)	Each axis can be independently powered down

It may be possible to use an "emulated" or "soft" gyroscope for lower-accuracy applications. Such a device isn't a gyroscope at all, but is a sensor-fused combination of an accelerometer and a magnetometer. We will return to this topic after discussing magnetometers.

Power-saving tips:

- Select a gyroscope designed for low-power use.
- Leverage power modes.
- Use FIFOs for batch reading.
- Use an accelerometer to bring the gyroscope out of sleep mode.
- Consider a soft gyroscope if your accuracy requirements are low.
- Consider combos (see below) if other sensors are needed in addition.

ambiq micro
Giving batteries a better life

Magnetometers

Magnetometers sense the orientation of a device within the earth's magnetic field. They can therefore measure the same fundamental degrees of freedom that the gyroscope senses. But, by doing so in a different way, they complement and help to correct gyroscope measurements.

The challenge with magnetometers is that they need to be sensitive, since the earth's field is relatively weak. They can therefore be easily overwhelmed by so-called "magnetic anomalies" – any big, dense, metallic item that can itself perturb the earth's field. A large boulder, an elevator, or even a magnetic desktop trinket can have an impact. Some will be stationary – like a big storage tank – while others may move – like a large truck.

The ability to reject anomalies is critical to reliable magnetometer usage. Here a gyroscope can act as a sensor-fused cross-check against the magnetometer. If the magnetometer is showing a change in orientation and the gyroscope isn't, then it's probably an anomaly. Good anomaly-rejection algorithms can be a competitive advantage for a magnetometer vendor.

There are numerous technologies from which magnetometers can be built. The most common type for consumer-grade applications are Hall Effect sensors, because they're small, inexpensive, and consume little power. AMR (anisotropic magnetoresistance) sensors are less common, but compete well in cost and power with Hall Effect sensors. Other technologies (GMR, MTJ, Lorentz Force) tend to be more expensive or power hungry, and they are less likely to fit the requirements of a wearable device.

Example device:

Device	Mode	Current	Notes
MEMSIC MMC3316xMT	Normal	160 µA (max)	
(3-axis)	Power-down	1 µA (max)	
AKM	Normal	10 mA (max)	
AK8963			
(3-axis)	Power-down	10 µA (max)	Sensor and digital interface can be independently powered down.

Power-saving tips:

- Use a Hall-Effect or an AMR magnetometer.
- Leverage power modes.
- Use FIFOs for batch reading if available (less common on pure magnetometers)
- Consider combos (see below) if other sensors are needed in addition.

ambiq micro
Giving batteries a better life

Soft or Emulated Gyroscope

When less orientation performance is required, a “soft gyro” can replace a true gyro with lower power and cost. Such a device is a combination of a magnetometer, which provides the basic orientation reading, and an accelerometer. The purpose of the accelerometer is to help correct against magnetic anomalies.

Because an accelerometer is a more indirect way to detect anomalies, the quality of the sensor fusion algorithms will determine how well they work. Look for one that demonstrates good magnetic anomaly rejection. Unless you have access to good algorithms, this is something you should buy rather than try to build on your own.

Example device:

Device	Mode	Current	Notes
Kionix KMX61G	Operating	450 µA (typ.)	
	Sleep	1 µA (typ.)	
mCube MC7010	Active	260 µA (typ.)	Max peak add ~2440 µA
	Standby	8 µA	Accel and Mag can be independently put into standby mode. Accel has stronger variation with V_{DD}.

An accelerometer and a magnetometer can also be fused together into an eCompass and can be purchased in a single package. But in the eCompass application, the accelerometer is used as tilt compensation for the magnetometer; its sensor fusion algorithms will be different from those used to create a soft gyro. Even though they both contain the same basic sensors, they cannot be interchanged.

Power-saving tips:

- Leverage power modes.
- Manage accelerometer and magnetometer power separately.
- Use FIFOs for batch reading.

ambiq micro
Giving batteries a better life

6-axis Combos (Accelerometer + Gyroscope); IMUs

Many manufacturers combine accelerometers and gyroscopes into a single package. Such devices can give complete position information (location and orientation, which can be expressed together in the form of "quaternions") and are sometimes referred to as inertial measurement units (IMUs).

In some such devices, not only can the raw data from each sensor be read, but the device will include a processor to fuse the raw data into quaternion form, simplifying downstream calculations. If you don't need this capability, then either you can choose a unit without it, or you may be able to disable the processing unit to save some power. This becomes a consideration for system optimization when deciding where to do your sensor fusion, as discussed below.

Such combo sensors will allow you to manage power modes independently for the accelerometers and gyroscopes (even by axis). This would allow you to use the accelerometer to wake the gyroscope as described above.

Example device:

Device	Mode	Current	Notes
InvenSense MPU-6000	Accel+Gyro +Processing	3.9 mA (typ.)	
	Accel+Gyro	3.8 mA (typ.)	
	Gyro +Processing	3.7 mA (typ.)	
	Gyro only	3.6 mA (typ.)	
	Accel only	500 µA (typ.)	
	Accel, low power mode	10-140 µA (typ.)	Depends on output data rate
ST ASM330LXH	Accel normal	245 µA (typ.)	
	Accel low power	65/115 µA (typ.)	Depends on output data rate
	Accel+Gyro normal	4.3 mA (typ.)	
	Accel+Gyro power down	6 µA (typ.)	Difference between "low power" and "power down" modes not clear

Power-saving tips:

- Leverage power modes.
- Manage accelerometer and gyroscope power separately.
- Use FIFOs for batch reading.

9-axis Combos (Accelerometer+Gyroscope+Magnetometer); IMUs

Taking combinations one step further, accelerometers, gyroscopes, and magnetometers can be co-packaged. These are referred to as having 9 axes even though only six degrees of freedom apply (the gyroscope and magnetometer both measure rotation). They're also referred to as IMUs.

With such a device, you can access any of the sensors individually, or their outputs may be combined and made available as quaternions. The number of modes may grow, since each sensor type may have its own power modes.

Example devices:

Device	Mode	Current	Notes
Bosch BMX055	Add accelerometer, gyroscope, and magnetometer currents below for total		
	Accelerometer		
	Normal	130 µA (typ.)	
	Suspend	2.1 µA (typ.)	
	Deep suspend	1 µA (typ.)	
	Low power 1	6.5 µA (typ.)	
	Low power 2	66 µA (typ.)	
	Standby	62 µA (typ.)	
	Gyroscope		
	Normal	5 mA (typ.)	
	Fast power-up	2.5 mA (typ.)	
	Suspend	25 µA (typ.)	
	Deep suspend	<5 µA (typ.)	

ambiq micro
Giving batteries a better life

Device	Mode	Current	Notes
Bosch	Magnetometer		
BMX055	Active, low-power preset	170 µA (typ.)	Peak current 20 mA
	Active, regular preset	0.5 mA (typ.)	Peak current 20 mA
	Active, enhanced regular preset	0.8 mA (typ.)	Peak current 20 mA
	Active, high-accuracy preset	4.9 mA (typ.)	Peak current 20 mA
	Suspend	1 µA (typ.)	
Analog Devices ADIS16400	Normal	70 mA (typ.)	
	Low-power	45 mA (typ.)	
	Sleep	600 µA (typ.)	

Power-saving tips:

- Leverage power modes.
- Manage accelerometer, gyroscope, and magnetometer power separately.
- Use FIFOs for batch reading.

Microphones

Microphones are of particular interest in wearables for use in voice activation. This is an “always-on” function: the microphone must always be listening for commands, which has implications for the architecture and power strategy.

Microphones intended for such a function are likely to have a low-performance power-down mode. The idea is that the microphone can be operated solely to detect voice, or perhaps sound that might be voice; once detected, then it can switch into standard performance mode for higher audio resolution.

Unlike most other sensors, microphones have two ways in which they contribute to the power budget. An obvious contribution comes from the microphone itself. But the audio signal must then be post-processed to extract meaning. For example, once awoken by a sound, processing will be required to determine if the sound is a voice, and, if so, then to decode the command. For greater security, more processing would be required in order to determine whether or not the voice is from an authorized person.

Microphones are also increasingly being used in pairs or even arrays for noise cancellation and audio "focusing." Each additional microphone adds its own power, but the computing required to resolve the multiple microphone streams into a single audio signal requires yet more power from the processor.

Audio processing algorithms can be substantial, and appropriate computing budget needs to be allocated to ensure that the audio subsystem can achieve its performance targets. Exactly what the cost of that processing will be will vary dramatically by algorithm and by processor. Performance and functional tradeoffs may be required in order to stay within budget.

Example devices:

Device	Mode	Current
InvenSense INMP421	Normal	650 µA (max)
	Sleep	50 µA (max)
Knowles SPH0641LM4H-1	Normal	700 µA (max)
	Sleep	80 µA (typ.)

Power-saving tips:

- Leverage power modes.
- When sleeping, minimize audio processing.
- Stage wake-up levels. Sound is a minimal level; voice is a higher level; identified voice higher yet; identified command is highest.

Pressure sensors

Given some power budget, power sensors may be worked into wearables. They can act as altimeters to help correct for elevation (z-direction) inaccuracies in motion sensors, or they might be used for other novel applications such as measuring force on different parts of the foot while running.

Their use in industrial applications is well established; consumer usage is less well developed. They come in many configurations:

- As differential sensors, which have two ports whose pressure difference is measured, or single-port sensors
- Results may be absolute or relative to some baseline
- Packaging will vary according to the environment in which the sensor is intended to operate.

Pressure sensors aren't particularly power-hungry, and yet they may have power-down modes.

ambiq micro
Giving batteries a better life

Example devices:

Device	Mode	Current
Bosch BMP180	Advanced resolution	32 µA (typ.)
	Ultra-high resolution	12 µA (typ.)
	High resolution	7 µA (typ.)
	Standard	5 µA (typ.)
	Ultra-low power	3 µA (typ.)
ST LPS25H	High resolution	25 µA (typ.)
	Low resolution	4 µA (typ.)
	Power-down	0.5 µA (typ.)

Power-saving tips:

- Leverage power modes.

Sensor Fusion and Algorithms; Sensor Hubs

We've seen examples above of sensor fusion – algorithms that combine the outputs of separate sensors into a single result. Combining accelerometer and gyroscope outputs into quaternions, letting gyroscopes and magnetometers correct each other, soft gyros (and eCompasses) – these are all examples of sensor fusion. Further sensor fusion might, for example, combine a pressure sensor with the motion sensors to get a better fix on altitude.

In an effort to offload the AP in smartphones, so-called "sensor hubs" have been devised for this purpose. The idea is that a lower-power microcontroller (MCU) can be used to fuse sensor streams, letting the AP sleep until truly needed.

But wearables don't have an AP, so a separate sensor hub does not make sense for this application. That said, processing elements such as we saw in some six-axis sensors above are actually sensor hubs housed along with the sensors themselves, with inputs for other external sensor data streams. There is a valid question as to whether such calculations are more efficiently done there vs. in the main MCU; we'll return to that question in the section on system optimization.

Regardless of where the fusion is performed, however, it's important to make sure that each reading is properly time-stamped, especially when batching from FIFOs. That's to ensure that readings from different sensors are properly aligned in time – or else, for example, you might fuse an accelerometer reading from one time with a gyroscope reading from another, giving a nonsensical position.

ambiq micro
Giving batteries a better life

Computing

The computing in a wearable device should be done by the smallest MCU that can handle the application. Simple in concept, it's actually no easy task to select an MCU, since there are innumerable combinations of processor, non-volatile and volatile memories, peripherals, and ports.

External memory also consumes power, although much less than what the other system elements might consume. In the computing subsystem, the MCU has by far the dominant effect on power.

Microcontrollers

With today's architectures, the MCU and communication (Bluetooth) consume roughly equivalent power in an overall system; an accelerometer will contribute less – roughly 60% or so what the MCU contributes. But radio and sensor chips are being re-architected to dramatically lower their power – as evidenced in the power modes discussed above. When those are used, the MCU becomes by far the dominant power consumer when standard MCUs are used.

This is the picture Ambiq is changing with their ultra-low-power Apollo MCUs. Because they're designed using so-called "subthreshold" techniques, they represent a complete departure from traditional MCUs, and can completely reset expectations of what power an MCU will consume, reducing it in some cases by an order of magnitude or more.

The following figure shows the impact both on systems using typical mainstream devices as well as the next generation. Using the Ambiq devices, MCU current drops from 30 µA to 4 µA – more than a factor of 7. Overall system current drops by 32% with older radio and sensor components; with newer devices, cutting the MCU current improves overall system current by 62%.

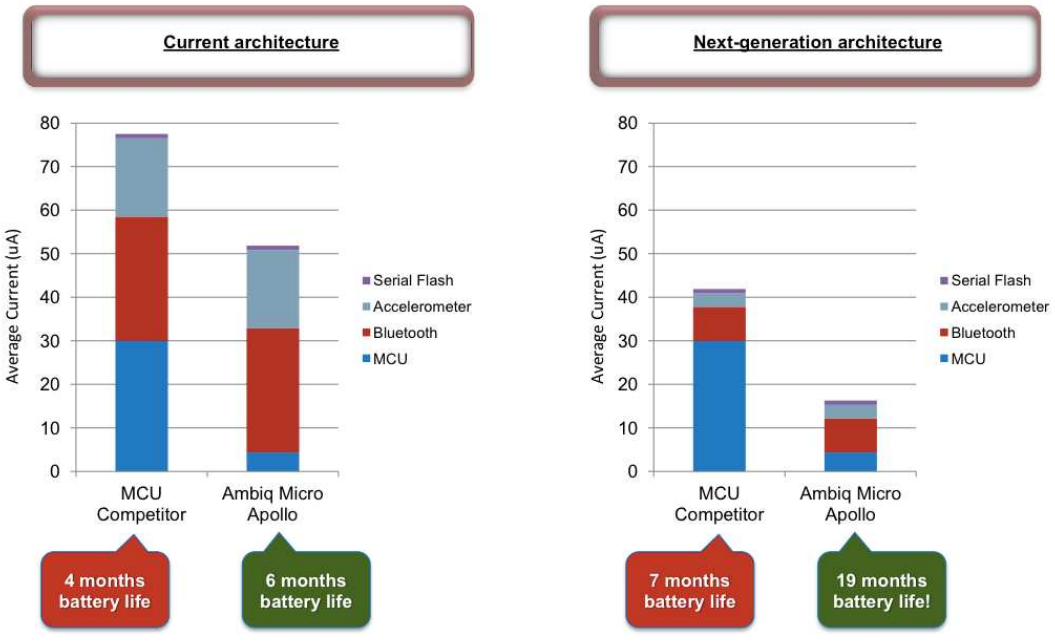

Figure 2: System current comparisons

ambiq micro
Giving batteries a better life

Datasheet values for supply current vary widely, depending on a range of parameters, and they can be complex to parse. Measurement conditions in the table below have been normalized as much as practical.

Example devices:

Notes on the following numbers:

- CoreMark algorithm unless otherwise noted
- Run from Flash (not SRAM) at 3.3V
- Peripherals disabled
- Power reduction techniques allowed as noted
- Typical numbers
- Sleep mode numbers assume timers are ON and no memory is retained

Device	Mode	Current	Notes
ST STM32F401 Cortex-M4F	Active	355 µA/MHz	Measurements performed with a reduced code that gives a consumption equivalent to CoreMark code. STMicro's adaptive real-time accelerator (ART) power reduction technique enabled.
	Sleep	2800nA	
Atmel SAM D20 Cortex-M0+	Active	99 µA/MHz	Uses a Cortex-M0+ which will be less efficient than a Cortex-M4F core.
	Sleep	3800nA	Uses Atmel's lowest-power "BACKUP" mode with 1KHz clock running.
Silicon Labs EFM32 "Zero" Cortex-M0+	Active	115 µA/MHz	Uses a Cortex-M0+ which will be less efficient than a Cortex-M4F core. Silicon Labs also does not publish CoreMark numbers in its datasheet – only prime number calculations are given (these result in a less accurate number since CoreMark performs a larger variety of tasks).
	Sleep	900nA	Uses EM2 "Deep Sleep Mode" with timers on
Ambiq Micro Apollo Cortex-M4F	Active	**35 µA/MHz**	As much as an order of magnitude lower in active mode power
	Sleep	**100nA**	As much as an order of magnitude lower in sleep mode power

ambiq micro
Giving batteries a better life

Multicore vs. Single Core

Embedded processors are increasingly available as multicore on the high end. Such processors are typically out of the range of what would be chosen for a wearable device. However, as processing demand and capabilities increase, it could be that multicore becomes an option.

If comparing a single-core and multicore MCU of the same computing capacity, there may be power advantages to selecting the multicore device – in particular if the device allows individual cores to be powered down. Then the full capacity can be turned on when needed, while shutting off extra cores when not needed. Careful analysis should be done, however, to confirm that, when running your actual code, the multicore unit will result in less power; you may not be able to tell by inspection.

That said, the smallest single-core MCU will consume less power than the smallest multicore MCU, so the multicore benefit applies only if you need that much processing power.

The other cost of multicore is complexity: depending on how the cores are utilized, it can be difficult to partition code between the cores. Issues like race conditions simply don't exist for single-core devices; they have to be considered in a multicore version.

In general, while there may be future opportunities for multicore use, single-core is far more likely in the near future.

Code Profiling

The actual code you run has an enormous impact on the power consumed. And different types of code will be more or less efficient on different MCUs. As a general rule, if you want to optimize your MCU selection, you will want to benchmark different MCUs against your code when selecting an architecture.

Given, however, that the Ambiq Micro Apollo MCU will consume less power than any other family, regardless of code, profiling becomes less important for selecting the device. Profiling will still be useful for understanding which code is causing the most consumption, allowing possible optimization through code changes, but at present, the Ambiq device will consume far less energy than any other commercially-available MCU.

The MCU architectural element having the biggest impact on power vs. code is the floating-point unit. Most small MCUs do not handle floating-point arithmetic, and so floating-point algorithms must be re-optimized using fixed-point math – a process that can take months. The Ambiq Micro Apollo MCU has a floating-point unit, making such re-optimization unnecessary – and with no power penalty as compared to other MCUs.

You may also be able to save power on certain common functions by executing them out of RAM instead of ROM. Use profiling to identify candidate code.

External Storage

External serial FLASH devices typically have low-power modes. Writing and erasing use more energy than reading; putting memories to sleep when possible will save power.

ambiq micro
Giving batteries a better life

Example devices:

Device	Mode	Current
Microchip SST25VF010A (1 Mbit)	Read	10 mA (max)
	Write/erase	30 mA (max)
	Standby	15 µA (max)
Macronix MX66L1G45G (1 Gbit)	Read	30 – 60 mA (max)
	Program	25 mA (max)
	Write status register	40 mA (max)
	Erase sector	25 mA (max)
	Erase chip	50 mA (max)
	Standby	200 µA (max)
	Deep power-down	40 µA (max)

Power-saving tips:

- Use the lowest-power technology that provides the needed performance.
- Leverage power modes in processors and in external memory.
- Profile code to identify which routines contribute most to power consumption.
- Run select code out of RAM.

Communication

Wearable devices overwhelmingly (if not exclusively) use the Bluetooth Low-Energy (BLE or BTLE) protocol to communicate with smartphones. The communication of data can require the same scale of power as the computing, so it's an important target for power reduction.

How much current is used for communication is a complex calculation, depending on a large number of variables. It may be worth building a model so that you can adjust different parameters to select your device and optimize how you use it. The following gives a general description of high-level factors, but actual calculations will depend highly on the chip you select and detailed timing.

ambiq micro
Giving batteries a better life

Basic Bluetooth LE States

While the original Bluetooth Classic protocol was optimized for streaming data, BLE has been designed for infrequent bursty data. This results in several states and has some non-intuitive power implications. In general, there are three high-level states for BLE:

- Active: transmitting and receiving data.
- Connected: paired up with a master (or slave), but not currently sending data.
- Unconnected: not paired up with a master (or slave).

(Note: because this discussion is an informal abstraction of the protocol details, it may use terminology differently from how it's used in the actual protocol. The intent here is to convey concepts at a higher level, without the burden of the details. You should always refer to the protocol for official specifics. Individual BLE chips will also have specific power states that will differ from what's presented in this higher-level discussion.)

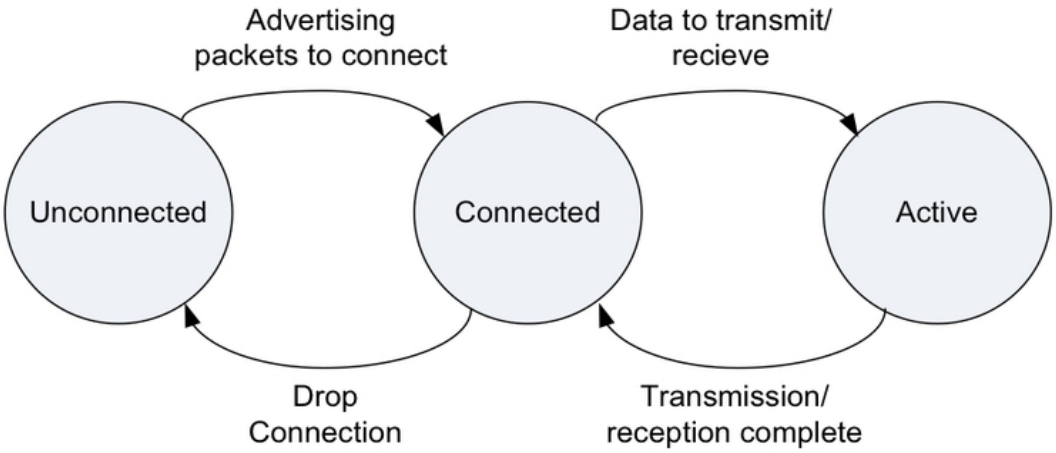

Figure 3: Abstracted Bluetooth Low-Energy states.

For bursty data with overhead to set up the bursts, it might seem obvious that power could be reduced by increasing the interval between bursts. But this may not be the case. In particular, it's not the case when in the connected state.

The difference between the connected and unconnected states is the existence of a pairing between a master and slave. There is no such pairing when unconnected, so increasing the length of unconnected time can reduce power.

However, while connected, the pairing must be maintained even if no data is being transmitted. This is done by having the master send a ping to the slave at a given interval, even if no data is being sent. The time between pings is referred to as the connection interval. While you might think that increasing the connection interval, making this ping less frequent, would decrease power, the opposite is true.

ambiq micro
Giving batteries a better life

In a wearable application, the wearable device is likely to be the slave. So the slave has to be ready to hear the ping from the master (assuming there is no data to transmit or receive). Even though the master and slave are conceptually paired, they're not electrically connected, meaning that they're running off of separate clocks. Even if the clocks are set to the same frequency, the physical clocks will differ from each other by some specified tolerance. So the longer the master and slave remain out of communication – that is, the longer the connection interval – the more the two clocks may have drifted apart.

The slave must awaken in advance of the master's ping so that it can be ready to receive it. The slave has to account for the fact that the master clock might be running faster, so the longer the connection interval – that is, the more time the master has had to get ahead – the earlier the slave has to wake up to be sure that it doesn't miss the ping. This effect is referred to as "window widening." The longer the connection interval, the wider the window that the slave must be awake to catch the ping. Because of that longer awake time, the slave can sleep less, and therefore more power is consumed.

The maximum connection interval is 30 seconds; beyond that, the connection must be dropped and re-established.

Going from unconnected to connected first involves the sending of advertising packets. Advertising packets are a new feature of BLE, and they can be used for a number of purposes, one of which is establishing a connection. A node wishing to set up a connection starts by sending out advertising packets; the other end of the connection can then respond and complete the connection.

For long intervals between data bursts or "heartbeat" packets, disconnecting between bursts will lower power consumption. For intermediate intervals, you should model whether it is lower power to remain connected, taking into account window widening, or disconnecting and expending the extra power required to advertise and reconnect. There will likely be some optimal point where data transmission is not too frequent and yet window widening doesn't dominate.

A complete model of BLE current consumption would need to include current consumed while transmitting and receiving data, while sleeping between pings, while responding to pings, during advertising, and when unconnected.

Depending on your BLE chip, you may have additional options for reducing current, including:

- Dynamic Window Limiting
- Application Latency
- Use of DC/DC converters or digital regulators

In addition, using a higher-tolerance external oscillator can reduce window widening. BLE chips typically have internal clocks, so this would mean the addition of an extra component.

BLE Chips

BLE chips generally come in two varieties: network processors or host controller interfaces (HCI). Both integrate the physical RF layer, but the HCI version implements only the lowest protocol layers, while the network processor implements the entire stack. Network processors may also be used in an HCI mode,

without implementing the upper layers of the stack. This becomes a consideration for deciding where to run the stack for minimal power consumption, which will be discussed below in the system optimization section.

You may also be able to specify the power level when transmitting data. The protocol itself has no provision for adjusting power according to detected signal levels, so a transmit power setting would either be permanent or would need to be optimized in real time based on custom code running above the BLE stack that implements some custom minimum power detection scheme.

Different chips will have different power states and state diagrams, and names like “idle” and “sleep” may mean different things with different chips. It’s important to study the datasheets carefully to understand what you’re comparing. In particular, take note of whether or not configurations and other data are retained when in low-power modes.

Example devices:

Device	Mode	Current	Notes
Nordic nRF8001 (Network Processor)	Receiving	11.1/14.6 mA (typ.)	Depends on DC/DC
	Transmitting	7 – 12.7 mA (typ.)	Depends on transmit power, DC/DC
	Standby	1.6/2 mA (typ.)	Depends on DC/DC
	Idle	2 µA (typ.)	
	Sleep	0.5 µA (typ.)	
Dialog DA14580 (Network processor)	Receiving	5.1/13.4 mA (typ.)	V_{DD}=1.2/3 V
	Transmitting	4.8/12.4 mA (typ.)	V_{DD}=1.2/3 V
	Sleep		Mode defined in datasheet, but no current value provided.
	Extended sleep	1.2 – 1.5 µA (typ.)	Depends on SysRAM and RetRAM
	Deep sleep	0.4-0.6 µA (typ.)	Depends on SysRAM, RetRAM, and oscillator configuration
ST STLBC01 (HCI)	Receiving	12.9 mA (typ.)	
	Transmitting	12.1 mA (typ.)	
	Idle	200 µA (typ.)	
	Sleep	19 µA (typ.)	
	Off	9 µA (typ.)	
	BLE Sleep	450 µA (typ.)	SPI transport layer only; Crystal
	BLE Sleep	60 µA (typ.)	SPI transport layer only; RC

ambiq micro
Giving batteries a better life

Power-saving tips:

- If the interval between bursts is long, disconnect between bursts. This is required if the interval is greater than 30 s.
- Model your data and timing to optimize timing parameters.
- Select the lowest transmit power possible for your application.
- Leverage other power modes when not exchanging data.
- Study datasheets for chip-specific power-saving features and recommendations. They vary by manufacturer.

System-Level Optimizations

The simple sense/compute/communicate model works at a high level, but further optimization may be possible. It's common for sensors to have some computing capability, and all BLE chips have an embedded microcontroller. So there may actually be some computing going on in all three blocks; the question is, can power be saved by moving code from the sensors or BLE chips to the central MCU?

At a generic level, any such code in question can be profiled to determine the power impact on either the native chip or the MCU. Depending on the specific chip, sensor, or MCU, it won't be obvious up front where the solution might land; profiling provides that information.

The picture changes with the Ambiq Micro Apollo MCU, since its power consumption is dramatically lower. It significantly tilts the balance in favor of saving power by moving calculations out of sensor and BLE chips into the Ambiq MCU. The following discussions dig into the specific calculations that make good candidates for moving.

Sensor Computations

A sensor typically consists of up to three elements:

- The sense element (often MEMS); a mechanical device that provides basic transduction into a raw electrical signal.
- Signal conditioning, which may remove noise, linearize, calibrate, and digitize the raw reading.
- Fusion, where signals from multiple sensors – both within the same package as well as from outside, via inputs provided for that purpose – can be fused into higher-level information. eCompass, soft gyro, and quaternion calculations are done here.

Not included in the discussion is control logic that manages the timing of sensing samples, delivery of digitized data into registers, FIFO management, and other such administrative tasks; such circuits are outside the sensor data path.

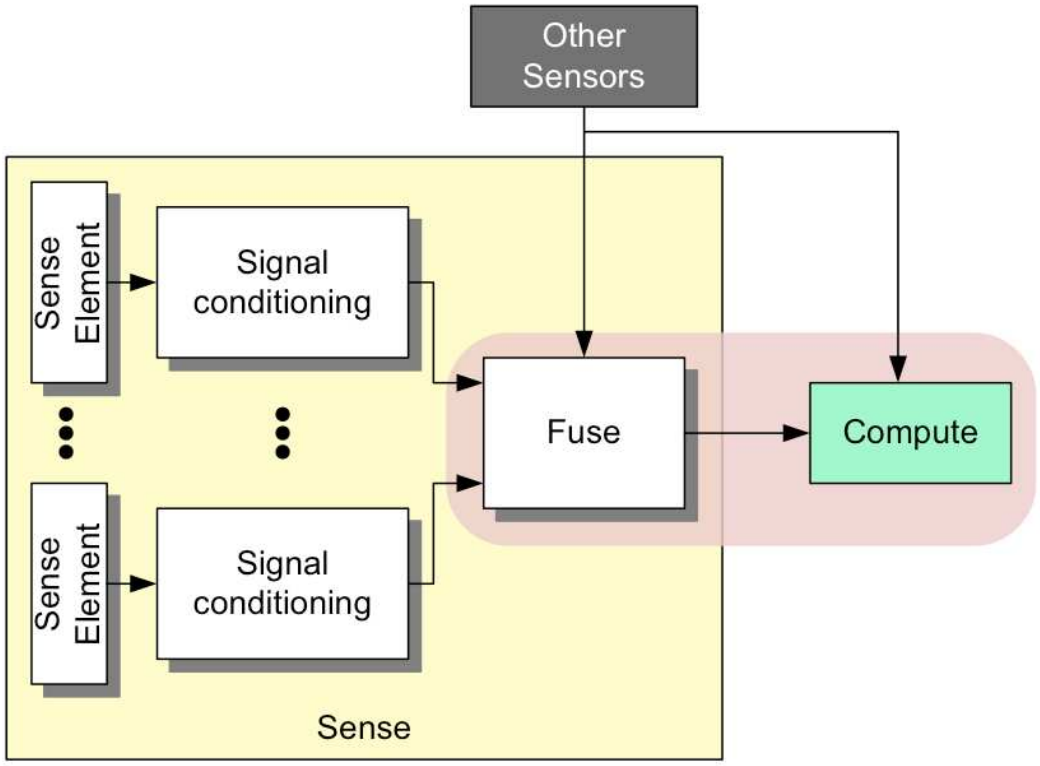

Figure 4: Sensor fusion code can be executed within some sensors or within the MCU

Signal conditioning is a very low-level operation, and it relies on calibration data taken during sensor testing. That calibration data is required for generating a reliable sensor signal, and the data is not available to outside circuits. So the signal conditioning block is not a candidate for moving to the MCU.

The fusion block, however, if it exists, is a candidate for moving to the MCU. In fact, the fusion is usually handled by a small MCU embedded in the sensor; inputs are often provided to allow other external sensor signals to be fed in so that one sensor acts both as sensor and sensor hub. Your options are:

- Fuse all in the sensor.
- Fuse only internal sensors in the sensor MCU, moving any fusion involving other sensors into the external MCU.
- Move all fusion to the external MCU, either selecting a sensor without an internal MCU or disabling the sensor's MCU to save some power.

If you move all of the fusion out of the sensor, then you'll need access to good fusion algorithms that you can implement on the external MCU. Such algorithms may be purchased, and open-source plain-vanilla algorithms are now becoming available through programs like the Accelerated Innovation Community.

Bluetooth Computations

As mentioned above, BLE chips can implement just the low-level portions of the stack (the HCI) or the full stack (network processor).

It's not practical to implement the HCI functions in the external MCU because there is no way to interface the external MCU directly to the RF circuitry. The upper layers of the stack, however, can be moved to the external MCU. Even though the BLE embedded MCU may still be executing, it will be doing less work and will therefore dissipate less power. Your options are:

- Use a network processor and process the entire stack in the BLE chip.
- Use a network processor, but only use the HCI portion of the BLE chip, handling the upper layers in the external MCU.
- Use an HCI chip, handling the upper layers in the external MCU.

If you do move the top-layer processing to the external MCU, you'll need access to code for implementing those stacks. Such code is widely available, often from MCU manufacturers, optimized for their MCU.

Summary

Wearable devices have, so far, focused on implementing a few simple functions – typically fitness-related. A critical barrier to more complex devices has been power consumption and the resulting short battery lives.

Power consumption is dominated by the sensors, microcontrollers, and BLE chips included with each wearable device. Power can be minimized both through careful component selection and by implementing the power-saving techniques discussed above.

Components intended for use in wearables are increasingly being designed with reduced power requirements; more improvements are expected over the next few years. The biggest change so far has been the availability of the Ambiq Micro Apollo microcontroller, which, based on its novel subthreshold design, consumes an order of magnitude less power than traditional MCUs.

Combined with power-stingy sensors and BLE chips, it will be possible to create increasingly sophisticated wearable electronics while maintaining or extending battery life.