GNSS Whitepaper
Location Services in IoT: Design Considerations for a Low-Power GNSS Receiver
Introduction

GNSS (Global Navigation Satellite System) constellations like GPS and others help us to find the easiest and fastest routes to our destinations, help us find our devices when they’re lost, and enable us to survey and map the pathways of our world. Until now, consumers have mainly used GNSS in their automobiles and mobile phones. Now there are emerging market opportunities for GNSS in the Internet of Things (IoT).

Within battery operated IoT devices such as remote IoT sensors and edge devices, wearables and health monitors, consumer mobile products, automotive after-sales products like insurance boxes and road tolling equipment, asset tracking devices, and others, Location Based Services (LBS) based on GNSS will have a major impact across our homes and cities. GNSS will be a key requirement going forward for IoT applications, many of which will mandate it for regulatory/insurance purposes.

Due to extreme power constraints, designing a GNSS receiver for a battery-operated IoT system is a daunting challenge compared to those in the systems we see in our vehicles and smartphones today. IoT edge devices can be untethered, needing to survive on either battery power or energy harvesting for weeks, months, or even years at a time without a battery re-charge or replacement. As such, the GNSS receiver inside these devices must have sub-milliwatt power consumption, versus the 20-30mW of traditional GNSS receivers. The industry needs a new approach to GNSS receiver design to make LBS a possibility for these new devices and applications.

Companies are looking to solve the power challenge in numerous ways, through hardware optimizations, software techniques, LTE assistance for improved time to first fix, new techniques to acquire location, and others.

In this paper, we will discuss design considerations and tradeoffs for a low-power GNSS receiver, including the main contributors to power consumption, and how you can design your receiver to provide the best solution possible within the power constraints of a battery-powered IoT device.

For the sake of this paper, we’ll focus on receivers for civilian bands used in consumer products (not military, maritime, emergency and other GNSS signal bands).

GNSS Receiver Basics

A typical GPS receiver system consists of a GNSS antenna, an RF front-end and a baseband signal processing engine. The antenna receives and amplifies the GNSS signals from the satellites, the RF front-end further amplifies the signal and digitizes it for processing, and then the baseband processor (also known as a correlator) decodes the digitized signals and performs a series of computationally intense signal processing operations to calculate position, velocity and time (PVT). The algorithms that convert the raw data into PVT information are part of the CPU, which can be either integrated with the baseband processor or operated separately.
Traditional GNSS receivers acquire, track, and decode GNSS navigation data in real-time, and acquiring/tracking the signal is the most complex and compute-intensive part of the process. For an accurate positioning calculation, the receiver must collect data and correlate code from at least four different satellites.

The signals that are received are often weak, since they must pass through the near-vacuum of space, then continue on through the various layers of the atmosphere to the earth. This means the receiver needs time to process the signal so that it can rise above the level of noise. This takes about 30 seconds – a time during which battery life drains from the device. Even in situations where the system is assisted in getting to first fix by LTE or Wi-Fi, six seconds are needed.

Research by The University of Texas has shown that baseband processing accounts for over half of the energy consumed in modern GNSS receivers. Since power consumption generally increases significantly during the time that the receiver is searching for and acquiring satellites, there are many techniques designed to keep power consumption to a minimum. Most hardware GNSS receivers feature sleep or standby modes during which the RF front end and baseband processor are powered down, leaving only the CPU on. The CPU wakes upon receiving a signal then activates the RF and GNSS circuitry. There are also more aggressive designs where even the CPU is powered down.

**The Constellation Matters**

Today there are four major satellite constellations. These include GPS (originating in the United States), GLONASS (Russia), Galileo (Europe), and BeiDou (China). In addition, there are numerous regional Satellite Based Augmentation Systems (SBAS) systems that help improve accuracy, including Wide Area Augmentation System (WAAS) and the European Geostationary Navigation Overlay Service (EGNOS).

While each of the four major constellations is designed around a similar GNSS topology that is comprised of space satellites, control network and user equipment, each was architectured in a different way and therefore building support for each constellation into the receiver has different power consumption implications.
There is one school of thought that GNSS consumes too much power in general to be used in battery operated IoT systems. To address this, companies are looking at other geolocation methods such as approaches for asset tracking using LWPANs.

But since the energy consumption of a GNSS receiver is often too high because it is tracking all four constellations in real-time, there are other approaches to controlling power consumption.

One possibility is to track only one or two of the constellations versus all four. While tracking all four will provide the most accurate solution (and most smartphone chips today support all four), not every application requires such pinpoint accuracy. If you’re designing an ultra-low power asset tracking device, it might not be necessary.

The two constellations which require the least power consumption in the baseband processor are GPS L1 and L2 C/A (for civilian use) and Glonass – which were the first two constellations in existence. While each of the constellations is designed to be ‘spread spectrum,’ the structure of the signals in the newer systems have different modulation techniques and signal parameters. Galileo and BeiDou were designed with larger signal bandwidth, more complex and longer codes, and higher navigation data rates. The aim is greater accuracy, but the changes have an impact on the receiver design since they require wider registers at higher operating frequencies.

Another challenge with tracking all four constellations is that some are on different frequencies and thus may require more than one radio. Only GPS and Galileo are on the same frequency.

Which constellation your receiver supports can also be dictated by more commercial concerns. For example, if your receiver will be integrated into a car that is imported to Russia, it’s mandated that the receiver support Glonass.

To ensure the most commercial and technical advantages, one option is to design support for all four constellations in hardware and make that support flexible to be programmed in software.

**Sampling Rate**

The sampling rate of a GNSS receiver, typically expressed in samples per second, or hertz (Hz), refers to the rate at which the analog signal is sampled in order to be converted into digital form. The sample rate determines how much processing is needed; in general, you can expect that 2x the sample rate leads to ~2x the processing. As you might expect, while a higher sampling rate leads to greater accuracy, it also leads to higher power consumption.

Which constellation(s) your receiver supports defines the sample rate. For example, the increased signal bandwidths of constellations like Galileo demand higher sampling frequencies. In keeping with their overall power consumption as outlined above, Glonass has the lowest sample rate and is the least complex to process, followed by GPS. Galileo is the most accurate, but its development has significantly increased the complexity of searching for a signal and tracking it—it’s 8x harder to find than GPS. BeiDou falls in the middle in terms of complexity.

**Processing requirements**

The complexity of the baseband processor is impacted by the number of bits that are needed to represent the intermediate signals, the bit-width of the registers, and the minimum frequency needed to process a particular signal.
Each of the different constellations has different resource requirements in terms of the number of registers and combinational logic. As we’ve seen, newer constellations have increased complexity (e.g. increased code lengths and chipping rates), leading to the need for higher clock frequencies and additional memory. Keeping bit widths as small as possible can help to optimize memory and power consumption. As with other parameters, there is a tradeoff between power and accuracy.

Software or Hardware?

A GNSS receiver can be built in hardware or software or mix of both. A software-based GNSS receiver replaces the functions of the baseband processor with software running on a general-purpose CPU, and the general advantage is significantly increased flexibility. In addition, with a software based GNSS receiver, it’s possible to update the product with new features and functionality throughout the product life cycle and thus extend the product’s lifetime. This isn’t possible with a purely hardware-based receiver.

There is also an argument that power consumption can be decreased by designing the receiver in software because there is no need for a baseband processor. But with traditional software-based receiver technology, a significant amount of digitized GNSS signal data must be processed by the CPU which in turn requires a higher-performance CPU running at higher frequency, leading to increased power consumption.

To reduce power, software-based receivers often use Fast Fourier Transform (FFT) algorithms which sample a signal and divide it into its frequency components, but this technique leads to increased memory requirements and thus higher cost.

Hardware based receivers offer better efficiency in terms of both power consumption and computational load. This is because a hardware design can be highly optimized and specifically designed for the sole purpose of GNSS processing.

We believe that the ideal solution is one that mixes hardware and software to provide a balance of flexibility and power.

Not Every Hardware Approach is Equal

The need to track the major GNSS constellations as well as SBAS leads to a dramatic increase in the complexity of tracking and positioning algorithms and firmware. Tracking multiple frequencies and signal types on each constellation also leads to an increase in the number of channels that must be implemented on the hardware to track the different satellites. This leads to an increase in cost, power consumption and silicon area.

However, there is a way to design the receiver for multiple satellites without such a resource-hungry multi-channel-based approach. For example, Imagination’s Ensigma GNSS IP, which is a hybrid hardware/software receiver, is based on a dedicated processing engine which is used elsewhere for Wi-Fi, Bluetooth, TV and radio, and thus uses more generalized processing techniques. It is designed it to be flexible, and to be reused again and again – running the data through the correlator multiple times, and time-slicing it for the different satellites.

Minimize your Radio Time

As discussed, the radio consumes a great deal of power when it’s turned on while it is tracking and acquiring satellites. Therefore, it’s important to turn the radio off as quickly as possible, and to keep power consumption below 10mW – ideally 5mW.

A lot of GNSS chipsets synchronize data in real time and therefore are continuously on. But not every application requires an always-on, real-time approach. There are many applications which only need to periodically deliver location data. Examples include use cases such as finding livestock in a field, pinpointing a streetlight that needs maintenance, determining driving patterns for insurance purposes, locating a stolen bicycle, and many others. Even wearable fitness monitors don’t need to deliver location data continuously. For such applications, it is possible to periodically turn on the radio for extremely brief periods of time.

Take a Snapshot

The idea of a ‘snapshot’ approach, also known as capture and process, is to quickly turn on the receiver, capture the satellite data, store it in memory, and then process that data offline after the receiver is powered down. Snapshot technology is designed to work with only a few milliseconds of a raw GNSS signal.

Snapshot technology has already been implemented in some industry software solutions. Imagination has taken snapshot to the next level by implementing it in its Ensigma GNSS IP. In the Ensigma hardware/software receiver implementation, the radio turns on for 100
or 200 milliseconds, captures the data at a very low sample rate (generally 1-bit), compresses the data and stores it in memory.

Once the 128-bit memory is filled, the radio can be turned off and the Ensigma engine begins to cycle the data through the correlator several times in fast succession to get a fix on the satellites.

The processing time is not limited by the speed of the data transmission; it’s limited only by the clock speed of the correlator processing. With optimized bit widths, the Ensigma hardware doesn’t need to work as hard as other hardware-based receivers, so it consumes less power. One way in which Imagination has worked around the accuracy tradeoff is by enabling bit width flexibility, for example searching for satellites on 1-bit, and then tracking satellites on a higher bit rate.

With the snapshot technique, there is no need to track satellites or decode navigation messages in real time, leading to dramatically reduced receiver power consumption. Since the GNSS receiver can be aggressively duty cycled, average power consumption can be further reduced.

In Summary
Designers of GNSS receivers for battery-operated IoT devices are under pressure to support the multiple broadcast frequencies needed for the different constellations; to make antennas smaller, lighter and lower cost; to guard against increasing amounts of interference in the L-band RF spectrum; and to meet aggressive low-power consumption targets. They must take into account the power consumption implications of each element of their GNSS receiver from the constellation choice to the baseband processor bit widths and memory sizes to the real-time (or non-real-time) requirements of their application. For many emerging battery-operated IoT devices, snapshot technology on a receiver that leverages both hardware and software can provide an ideal ultra-low power GNSS receiver solution.