SEMTECH

Migrating an Internet of Things (IoT) Sensor Design to LoRaWAN

LoRa®

www.semtech.com

Semtech White Paper

Migrating an Internet of Things (IoT) Sensor Design to LoRaWAN

INTRODUCTION

The success of Semtech's LoRa® devices and wireless radio frequency (RF) technology (LoRa Technology) in Low Power Wide Area Network (LPWAN) IoT applications speaks for itself: over 70 announced LoRaWAN public operators in 100 different countries as of January 2018 with an ecosystem that is supported by more than 500 members of the LoRa Alliance™ (www.lora-alliance.org) whose mission is to standardize LoRaWANs. The LoRa Alliance is an open, non-profit association of members consisting of a variety of different companies across many different sectors including semiconductor companies, network operators, sensor manufacturers, software, and hardware manufacturers. LoRa Alliance members collaborate to drive the global success of the LoRaWAN™ protocol.

LoRaWAN is an ideal Internet of Things (IoT) protocol for low data rate, low power, low cost, and long range sensor applications in a variety of vertical markets. Today, there are millions of LoRaWAN end-nodes connected to thousands of gateways around the world. This paper provides a high-level overview of Semtech's LoRa modulation technology, the LoRaWAN open protocol and network topology, and discusses the major steps necessary to convert an existing sensor design to be compatible with a LoRaWAN network.

WHAT IS LoRa®?

LoRa (short for long range) is a proprietary spread spectrum modulation technique derived from existing Chirp Spread Spectrum (CSS) technology. It offers a trade-off of sensitivity versus data rate while operating in a fixed bandwidth channel of either 125Khz or 500Khz for uplink channels and 500Khz for downlink channels. It uses orthogonal spreading factors which allow the network to make adaptive optimizations of an individual end-node's power levels and data rates with a goal of preserving end-node battery life.

For example, a sensor located close to a gateway should be transmitted at a low spreading factor since very little link budget is needed. A sensor located several miles from a gateway will need to transmit with a much higher spreading factor as the increased spreading factor will provide increased processing gain resulting in higher RX sensitivity, but at a lower data rate.

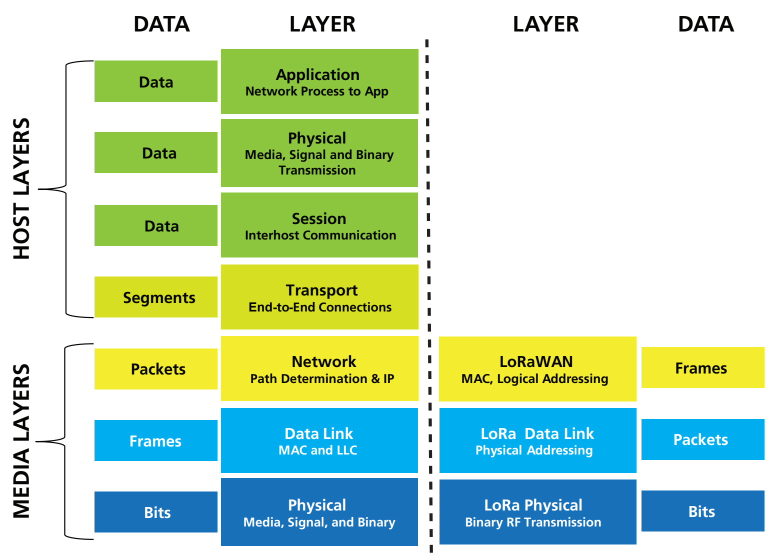

Figure 1: OSI 7-Layer Network Model

LoRa itself is purely a PHY (Bits) layer implementation as defined by the OSI 7-Layer Network Model in Figure 1. Instead of a cable, the air is used as a medium to transport the LoRa radio waves from an RF transmitter in an IoT device to a RF receiver in a gateway and vice versa.

In a traditional or Direct Sequence Spread Spectrum (DSSS) system, the carrier phase of the transmitter signal changes according to a code sequence as shown in Figure 2. When multiplying the data signal with a pre-defined bit pattern at a much higher rate, also known as a spreading code (or chip sequence), a "faster" signal is created that has higher frequency components than the original data signal and, as such, spreads the signal bandwidth beyond the bandwidth of the original signal. In RF terminology, the bits of the code sequence are called chips, in order to distinguish between the longer un-coded bits of the original data signal. When the transmitted signal arrives at the RF receiver, it is multiplied with an identical copy of the spreading code used in the RF transmitter, resulting in a replica of the original data signal.

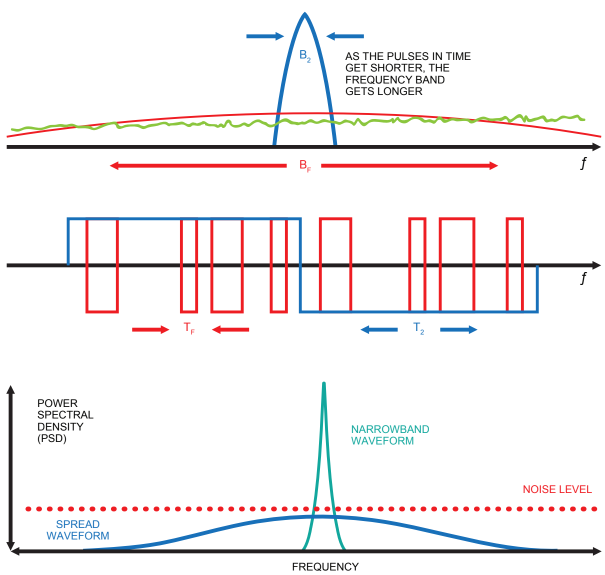

Figure 2: Example of Direct Sequence Spread, Modulation

Why go through all this trouble you may ask? Why not just transmit the original data signal instead of going through this code sequence multiplication. The Log10 ratio of the code sequence's chip rate and the data signal's bit rate is called the processing gain (Gp). This gain is what allows the receiver to recover the original data signal even if the channel has a negative Signal to Noise Ratio (SNR). LoRa has a superior Gp compared to frequency-shift keying (FSK) modulation, allowing for a reduced transmitter output power level while maintaining the same signal data rate and similar link budget.

One of the downsides of a DSSS system is the fact that it requires a highly accurate (and expensive) reference clock. Semtech's LoRa Chirp Spread Spectrum (CSS) technology offers a low cost and low power, yet robust, alternative to a DSSS system that does not require a highly accurate reference clock. In LoRa modulation, the spreading of the signal's spectrum is achieved by generating a chirp signal that continuously varies in frequency as is depicted in Figure 3. An advantage of this method is that the timing and frequency offsets between transmitter and receiver are equivalent, greatly reducing the complexity of the receiver design. The frequency bandwidth of this chirp is equivalent to the spectral bandwidth of the signal. The data signal that carries the sensor data is chipped at a higher data rate and modulated onto the chirp carrier signal. LoRa modulation also includes a variable error correction scheme that improves the robustness of the transmitted signal. For every 4 bits of information sent, a 5th bit of parity information is sent.

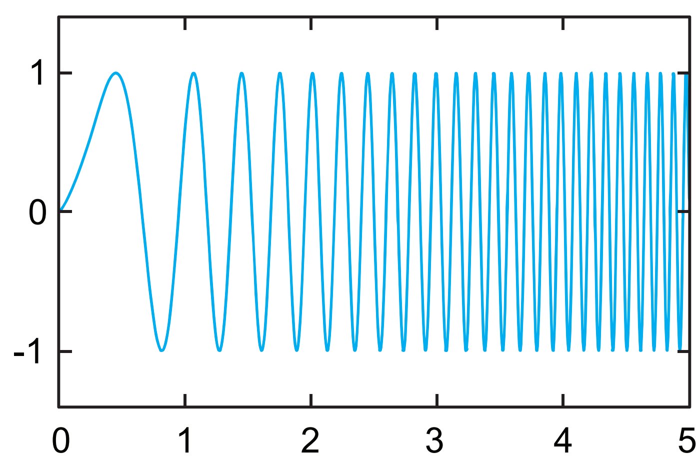

Figure 3: LoRa Chirp Signal Example

KEY LoRa MODULATION PROPERTIES

As discussed in the previous paragraph, LoRa processing gain is introduced in the RF channel by multiplying the data signal with a spreading code or chip sequence. By increasing the chip rate, one increases the frequency components of the total signal spectrum. In other words, the energy of the total signal is now spread among a wider range of frequencies, allowing the receiver to discern a signal with a worse SNR. In LoRa terms, the amount of spreading code applied to the original data signal is called the spreading factor (SF). There are a total of six spreading factors defined in LoRa Modulation: [SF7..SF12]. A signal modulated with a larger spreading factor will be able to travel a longer distance and still be received without errors by the RF receiver compared to a signal with a lower spreading factor.

Spreading Factor (for UL at 125Khz)	Bit Rate	Range (dependent on terrain conditions)	Time on Air for an 11-byte payload
SF10	980 bps	8 km	371 ms
SF9	1760 bps	6 km	185 ms
SF8	3125 bps	4 km	103 ms
SF7	5470 bps	2 km	61 ms

Table 1: LoRaWAN Spreading Factors With TOA for an 11-byte Payload

Table 1 shows the four different spreading factors [SF7..SF10] that can be used for uplink (UL) messages on a 125KHz channel (Downlink messages use 500KHz channels that can use all six available spreading factors [SF7..SF12]. It shows the equivalent bit rate as well as the estimated range (this depends on the terrain; longer distances will be achieved in a rural environment compared to an urban environment). It also shows the time-on-air (TOA) values for a payload for each of the four spreading factors.

An inherent property of the LoRa modulation spreading factors is that they are orthogonal. This means that signals modulated with different spreading factors and transmitted on the same frequency channel at the same time do not interfere with each other. The Semtech SX1301 baseband processor (modem chip) can process up to six signals each with a different spreading factor coming in on the same channel concurrently because each signal appears as noise to the other.

LoRa signals are very resistant to both in-band and out-of-band interference mechanisms. LoRa modulation also offers immunity to multipath and fading, making it ideal for use in urban and suburban environments, where both mechanisms dominate. Doppler shifts cause a small frequency shift in the LoRa pulse which introduces a relatively negligible shift in the time axis of the baseband signal. This frequency offset tolerance mitigates the requirement for tight tolerance reference clock sources and, therefore, makes LoRa ideal for data communications from devices/sensors that are mobile.

THE RF LINK BUDGET OF A WIRELESS CONNECTION

The link budget of a wireless connection is the sum of all the gains and losses in dB from the transmitter, through the propagation channel, to the targeted receiver: Transmitter (TX) Gain, TX Antenna Gain, Channel Loss, and Receiver Antenna Gain. If this total sum is less than what is needed to meet the SNR at the receiver, the link cannot be established.

The easiest way to be able to meet the required link budget for a certain wireless connection is to increase the output power. However, in the RF world, TX output power is limited by the U.S. Federal Communications Commission (FCC), even when transmitting in a license free spectrum. LoRa RF transmitters operate in the ISM (Industrial, Scientific and Medical) Band. The maximum output power allowed is 30dBm (1 Watt) under a 50-channel frequency hopping scenario. Besides limiting the maximum output power of a RF transmitter, the FCC also sets a limit for the duration in time that this transmitter is turned on, called the dwell time. It limits the time that an RF transmitter operating in a 125KHz channel can be turned on to 400ms (0.4s). In the aforementioned 50-channel frequency hopping case, the RF transceiver (radio) chip is supposed to hop from one channel to the next in a random fashion, not to transmit on the same channel until it has cycled through all other 50 or more channels. In 8-, 16-, 32 channel -frequency hopping cases, the maximum allowed output power level is 20dBm (100mW). LoRa Modulation gives us an alternative to increasing the TX power; choosing a spreading factor with a higher coding gain. The higher coding gain provides a higher sensitivity on the receiver side, thus requiring less power on the TX side. LoRaWAN employs a battery saving Adaptive Rate Scheme (ADR) that trades TX output power versus spreading factor based on distance between each sensor and its closest gateway.

LoRaWAN: AN LPWAN NETWORK PROTOCOL

As discussed above, LoRa is a PHY (Layer 1 as shown in Figure 1) Layer protocol. A basic Layer 2 (Link Layer) packet structure has been defined for LoRa. As can be seen in Figure 4, this packet structure contains a Preamble (fixed bit sequence for the receiver modem to lock on to), a Packet Header, a Packet Header cyclic redundancy check (CRC), a Packet Payload, and an optional additional CRC. This Link Layer packet structure is defined by Semtech's LoRa Transceivers. It is possible to use these transceivers in point-to-point LoRa RF connections. A solution like this is a LoRa solution, but not LoRaWAN. It is important to understand the difference between LoRa and LoRaWAN. One can develop a product using a Semtech LoRa RF Transceiver using the orange packet structure as defined in Figure 4. It is left up to the developer how to use the PHYPayload; one has the freedom and flexibility to define their own protocol and use this PHYPayload for a combination of commands and user data. Any custom implementation like this is referred to as a LoRa solution.

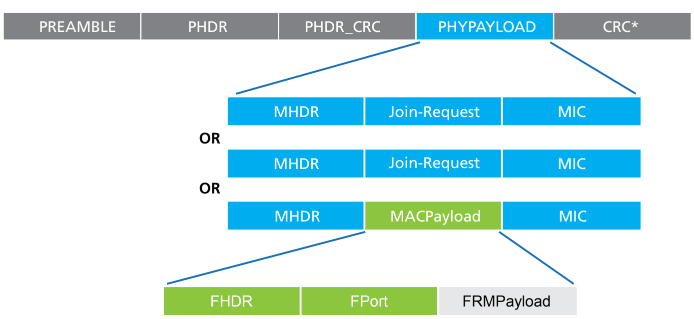

Figure 4: LoRa PHY Layer Packet Structure and LoRaWAN MAC and Frame Definitions (LoRaWAN 1.0.2 specification)

Companies that have developed existing LoRa-based (but not LoRaWAN) solutions should, in most cases, be able to convert their LoRa design to LoRaWAN without any hardware (HW) changes since the RF chip or module in their design does not change. They will have to convert their existing proprietary protocol to the LoRaWAN protocol which will require a new protocol stack. As long as the LoRaWAN protocol stack (< 50Kbytes) will fit in their existing controller memory space, migrating to support LoRaWAN will only involve software (SW) development. When migrating from a proprietary protocol to the LoRaWAN protocol, one must pay close attention to protocol stack code space requirements, and also to the user payload size and make adjustments as needed to be able to fit the existing data into the LoRaWAN payload such as basic data compression.

LoRaWAN is a Layer 3 (see Figure 1) Network Layer Protocol that is carried in the PHYPayload section of the LoRa PHY Layer Radio Packet. LoRaWAN is a LPWAN protocol specifically designed for low power, long range sensor applications. This LoRaWAN protocol provides the ability to create a high-capacity star network, and not just individual point-to-point LoRa links between two devices. There are different kinds of networks: 1) a real network where end-nodes or IoT devices can be connected by sending an over the air activation (OOTA) request to one or more LoRaWAN gateways, and 2) a network where there is a central entity in the Cloud that manages the network and optimizes individual end-node battery life by making trade-offs between end-node transmit power and spreading factors, etc. Finally, the protocol offers a network where the end-node's data is routed to the correct application server in the Cloud. The next paragraph will discuss the LoRaWAN network architecture.

Let us focus again on the diagram in Figure 4. This can be very confusing for someone that is not familiar with the 7-Layer OSI Model in Figure 1 or with networking in general, such as, Ethernet frames and IP packets. The goal of Figure 4 is to depict that there is a LoRa Packet Layer defined (In Figure 1, this can be viewed as the blue data link layer in the OSI model). This is the layer above the dark blue physical layer or the RF link in this case). The LoRaWAN protocol is transported inside the payload of the LoRa packets, just like IP packets are transported inside the payload of Ethernet frames. They have defined a MAC Header (MHDR), MAC payload and a Message Integrity Code. There are actually three different MAC structures as shown in Figure 4. Finally, the MAC payload contains a frame header, a frame port and the actual payload data (light grey). Anyone that is interested in converting a connected device/ sensor to the LoRaWAN protocol should pay close attention to this payload. One of the key questions to answer in a sensor conversion decision is, "Can I fit the sensor data of my existing IoT device into this LoRaWAN payload?" The LoRaWAN usable payload size depends on the channel frequency as well as on the spreading factor and varies from 11 to 242 bytes max (this is the range in maximum payload size in North America. Please refer to the regional parameters document of the LoRaWAN specification which details a variety of LoRaWAN specs for each specific region in the world). Our section on LoRaWAN Channel Plans and Data Rates provides more details.

For more information on the LoRaWAN protocol itself, as well as the regional parameters document, please download the latest versions from the LoRa Alliance's website at: www.lora-alliance.org as many of those details are beyond the scope of this white paper.

LoRaWAN NETWORK ARCHITECTURE

A LoRaWAN network is always implemented in a star topology as shown in Figure 5. Unlike a mesh topology, a star topology is ideal for power constrained (i.e., battery operated) end-nodes (sensors) because it only has to transmit its own messages and does not have to waste battery life relaying messages from surrounding end-nodes. A LoRaWAN network consists of one or more LoRaWAN gateways that are all connected to one central network coordinator, or so called network server (NS).

Unlike cellular base stations which have a high level of hardware and software complexity, and therefore a high cost, LoRaWAN gateways are basic protocol bridges with a much lower cost point. Each gateway receives LoRa modulated radio messages from all LoRaWAN end-nodes within radio distance. Every received LoRaWAN frame with a correct CRC code will be forwarded to the NS encapsulated in an IP frame. Gateways can be connected to the NS over Wi-Fi, hard-wired Ethernet or even a cellular connection. The gateway is, in essence, a bridge between LoRaWAN and IP. A managed LoRaWAN network (such as the Comcast's machineQ™ network) typically consists of both macro (64 channels) and picocell (8 channels) gateways. The macro cell solutions provide city wide (broad) coverage while the picocell gateways allow for increased network capacity in dense areas. An increase in the number of gateways in a specific area will typically prolong LoRaWAN end-node battery life in that same area as the distance between end-nodes and gateways is shorter. This will in turn allow LoRa radio packets

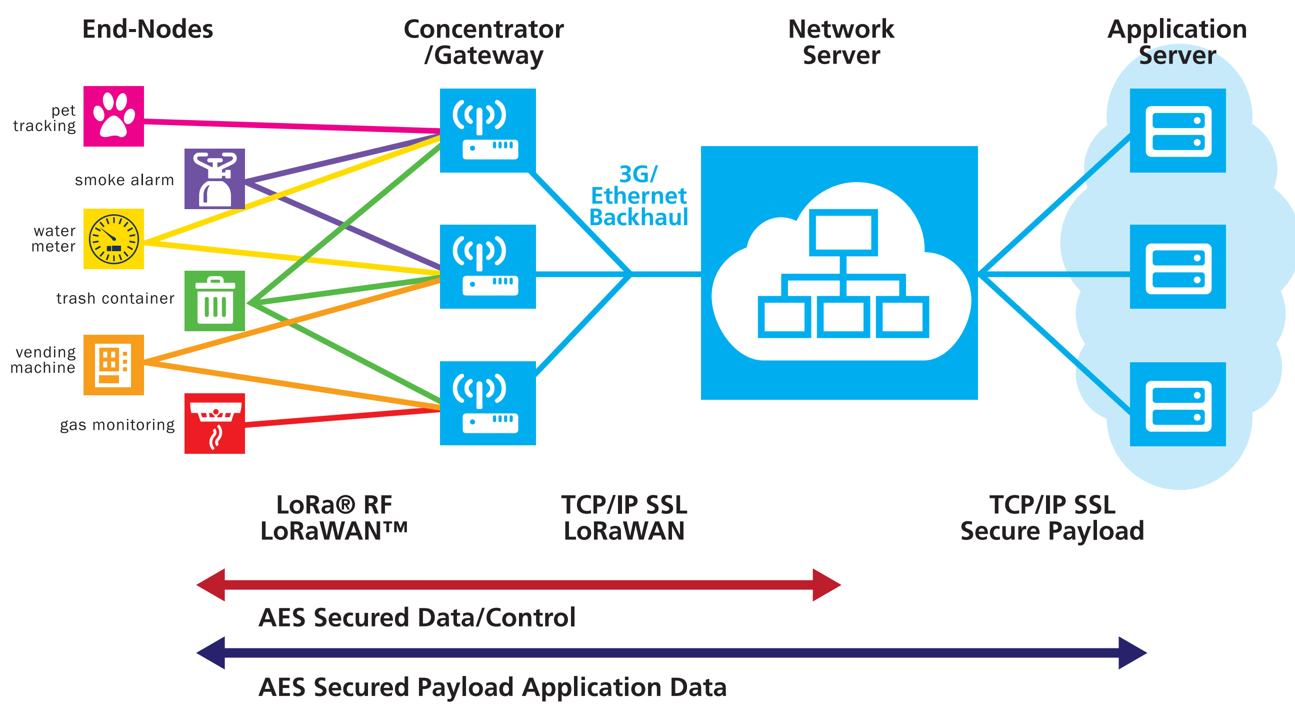

Figure 5: LoRaWAN Star Network Topology

to be transmitted with lower spreading factors which require shorter time on air. For example, a LoRaWAN frame with an 11-byte payload transmitted in a LoRa radio packet with SF7 has a TOA (Time on Air) of 61ms, which that same payload would require 371ms when transmitted at SF10 (as can be seen from Table 1). Another benefit of densifying a LoRaWAN network with picocell gateways is the ability to assure coverage in hard to reach areas such as basements.

In the network architecture pictured in Figure 5, the NS actively manages the LoRaWAN network. It coordinates incoming messages (LoRaWAN frames) from all the different end-nodes as well as network commands (so called MAC frames) between all the gateways and the application servers. Below is a list of the major management functions implemented in the NS:

- **IoT device over-the-Air Activation (OOTA).** The NS manages all requests from end-nodes to access the network (so called Join Requests), it informs each end-node which set of 8, 16 or all 64 channels to operate on, what spreading factor (SF) to use and what power level to transmit at.
- **Data de-duplication.** The NS deletes duplicate radio messages received from the same IoT devices from different gateways.
- **Dynamic DL LoRaWAN frame routing.** Selects the gateway best suited to connect with each LoRaWAN end-node to receive data link (DL) messages.
- **Adaptive Rate Control (ADR).** The main goal of ADR is saving the battery power of the LoRaWAN end-nodes. By having the end-nodes closest to a gateway transmit using the lowest SF, their time on air is minimized, as such prolonging their battery life. More distant sensors transmit at a higher SF. A tradeoff is made between battery power and distance as a higher SF allows for a gateway to connect to a sensor further away.
- **Network congestion.** The NS can instruct individual end-nodes to connect to a different gateway by changing their channel plan(s).
- **Forwards all application data to the right application server**. When a LoRaWAN end-node requests to join the network using a join request MAC command, the end-node transmits a specific application code (AppEUI) that is a unique code indicating which application server its data needs to be forwarded to. Once the end-node receives a join accept MAC command, all further frames containing user (application) data will be delivered to its designated application server.
- **Administration**. General network administration, network provisioning and reporting functions.

The NS has control over key operational and network settings in each LoRaWAN end-node: Transmit Power Level, SF, Channel Plan (i.e. which set or sets of 8 channels out of the available 64 available 125KHz uplink channels), receive window timing, etc. Control over the above mentioned parameters provides the NS the ability to optimize both network performance and end-node battery life. Critics might argue that one of the downsides of a star topology is that when the NS goes down, no data traffic is possible. A real-world managed LoRaWAN network implementation such as Comcast's machineQ LoRaWAN network would of course have a redundant NS implementation to cover such a scenario.

LoRaWAN SECURITY AND DEVICE AUTHENTICATION (LORAWAN 1.0.2 SPECIFICATION)

A key element of the LoRaWAN network protocol is security. Its baseline authentication and security framework is based on the AES 128 encryption scheme as implemented by IEEE 802.15.4/2006 Annex B [IEEE802154]. By using separate keys for user data encryption and authentication/network integrity, LoRaWAN offers a higher-level of security compared to single key implementations.

Figure 5 shows the two layers of network security: one for control data between the end-nodes and the NS (authentication and integrity) and one for the user data that is transported from the end-node to the application server (and back). As such, there are two specific 128-bit AES encrypted keys defined, the network session key (NwkSKey) and the application session key (AppSKey). Each IoT device will have its own unique NwkSKey and AppSKey.

There are two methods for an IoT device or so called end-node to be connected to a LoRaWAN network. The first one is called Activation by Personalization (ABP). In this method, the NwksKey and the AppsKey are already stored in the IoT device together with a unique 32-bit device address and a unique 24-bit network ID that identifies the specific LoRaWAN network the device is meant to connect to.

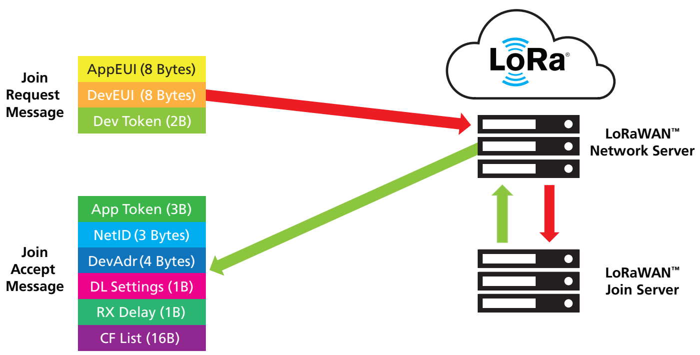

Figure 6: Join Request and Join Accept Message Formats

The default method of connecting a LoRaWAN IoT device is through a procedure called OOTA. In this method, each IoT device will send out a join request message to the NS which then forwards this message to a join server as shown in Figure 6. This join request MAC command will contain three data fields (see Figure 6). An IEEE defined 64-bit DevEUI (think of it as an "Ethernet MAC" address for LoRa devices) uniquely identifies this specific LoRaWAN end-node. It will also send a unique AppEUI identifying the application server this specific end-node wants to connect to. The final data is a random 2-byte device token. The join server will store these random device tokens from previous join requests messages from each end-node. If the join server receives a future join request from a specific end-node with a device token identical to a recently received one, the join server will ignore the join request. This will prevent so called "replay attacks" where a hacker could somehow capture a join request radio message from a particular end-node and replay (i.e. re-transmit) the same message with the intent to get the original end-device disconnected from the LoRaWAN network.

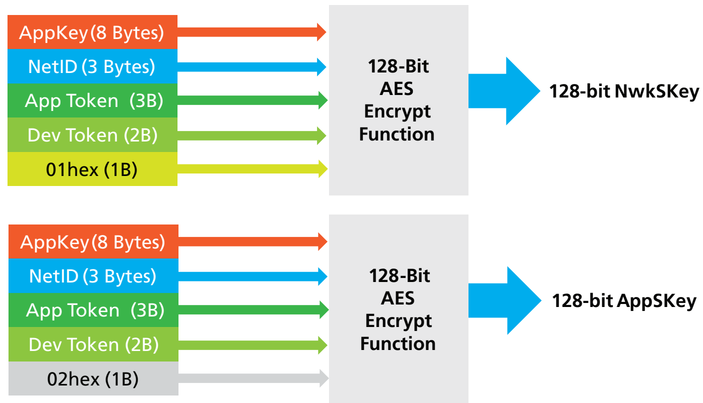

Figure 7: NwkSKey and AppSKey Generation (LoRaWAN 1.02 specification)

Only if the join server is able to authenticate the combination of DevEUI and AppEUI will it then issue a unique 32-bit device address, a unique 24-bit Network ID and a 3-byte application token. These parameters will be received by the end-node in a so-called join response command from the NS as shown in Figure 5. The end-node can then generate its own security keys. Figure 6 shows what fields are used to generate both the AppSKey and NwkSKey. One of the key fields to generate these keys is the AppKey. This is a unique 128-bit fixed value that is unique for each end-node.

Besides the above mentioned fields, the join response message will also inform the LoRaWAN end-node which channel plan to receive and transmit, and will also provide it with a few other RF provisioning parameters.

Each time the end-node loses network connectivity and or power, it will try to re-activate itself by transmitting a new join request message.

LoRaWAN END NODE CLASSES

LoRaWAN defines three different end-node types or classes:

Class A: Battery powered end-nodes that follow the "Aloha Protocol," meaning they transmit an uplink (message from the end-node to the NS) at a completely random time, i.e. there is no pre-set schedule or assigned timing slot. For every LoRaWAN uplink message, the end-node expects a downlink message (from the NS to the end-node). This downlink message can come in one of two receive downlink slots as shown in the timing diagram below in Figure 7. If the end node receives a DL in the first slot, it will not listen for the second receive time slot and power down its RX receiver.

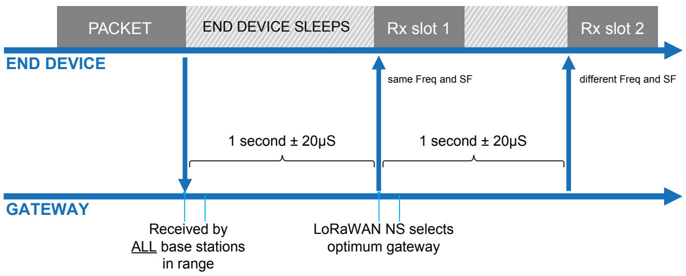

Figure 7: LoRaWAN End-node Class A Timing Diagram

After the DL message is received, the end-node goes back to sleep to conserve battery power and there is no procedure for the LoRaWAN NS to wake up the end-node. Class A end-nodes are a great choice for sensors, but not for actuators such as irrigation valves or door locks. The NS needs to be able to communicate with these types of end-nodes when there is an immediate need. For this purpose Class B sensors were defined.

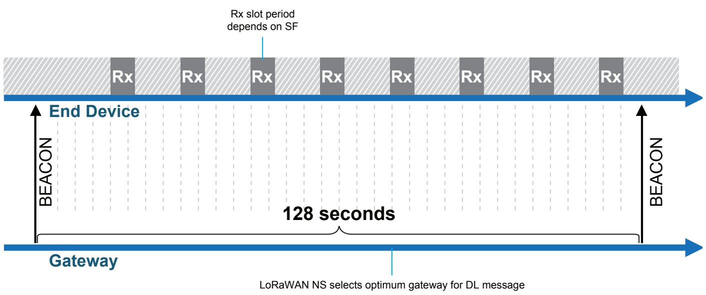

Figure 8: LoRaWAN End-node Class B Timing Diagram (only RX time slots shown)

Class B: Battery powered end-nodes that have been assigned a time slot based on a periodic beacon coming from the gateway. UL messages can be sent anytime the device is not supposed to be listening for a DL message. The end-node will wake up at a preset interval and listen for a DL message from the NS. End-nodes that need guaranteed periodic access from the network, such as wireless locks, sprinklers and shut-off valves, will require Class B support. Figure 8 depicts the timing diagram for Class B end-nodes. Class B devices will power up and join the network as a Class A device and can then request the NS to change operation to a Class B device.

Class C: Main powered end-nodes and always listening for a DL message from the NS, except for when they are transmitting an UL message. Class C devices are typically street light controllers, electric meters or, in general, any type of end-node that is connected to main power. Figure 9 depicts the timing diagram for Class C devices. Class C devices implement the same two receive timing windows as Class A devices, but unlike Class A devices, they do not close their RX2 window until they need to transmit again.

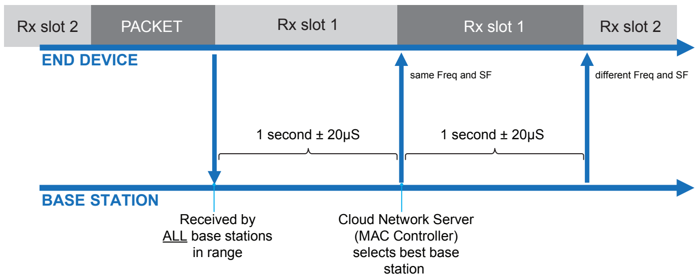

Figure 9: LoRaWAN End-node Class C Timing Diagram

All LoRaWAN IoT end-nodes have to at least support Class A. There is a possible scenario where IoT end-nodes configured as Class A or Class B can be temporarily (for a very short time) configured as a Class C device to always be listening. The LoRa Alliance technical committee is currently in the process of finishing a Firmware over the Air (FOTA) update procedure that will require all end-nodes needing a firmware (FW) update to temporarily switch to Class C mode where they will all listen at the same time to a sequence of multicast LoRaWAN FW update frames. The firmware file will be segmented in small fragments and parity information will be added to each segment. All end-nodes that do not receive all the fragments that make up the FW file will be able to rebuild the complete FW file by using the parity information contained in each of the fragments that they did receive. This mechanism is similar to a RAID disk array that allows the RAID controller to rebuild the data from a failed disk drive using the striped parity data that is stored across all the disk drives that make up the disk array.

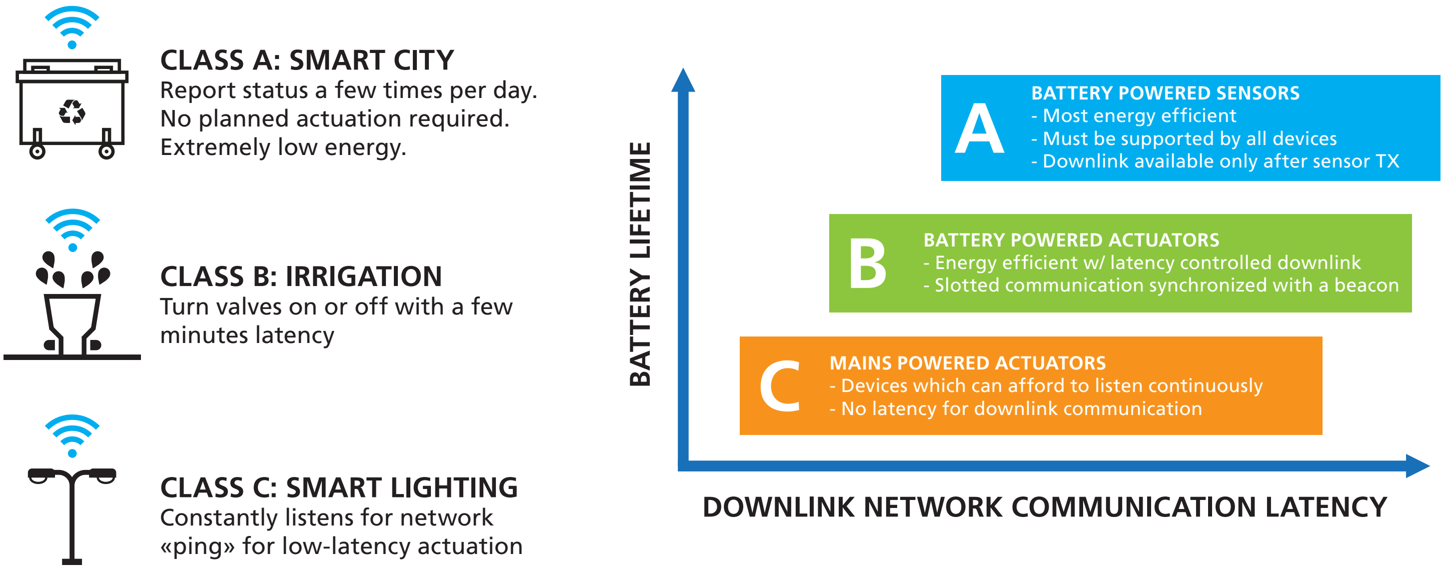

Figure 10: LoRaWAN End-node Classes

LoRaWAN CHANNEL PLAN AND DATA RATES

As mentioned previously, LoRaWAN operates in the ISM band. In North America, there are 64, 125Khz LoRa uplink channels defined, centered on a 200KHz raster as can be seen in Figure 11. There are eight 500KHz uplink channels as well as eight, 500KHz downlink channels defined.

In North America, gateways can have up to 64, 125Khz uplink channels as well as eight 500KHz uplink and downlink channels. This type of gateway is referred to as a carrier-

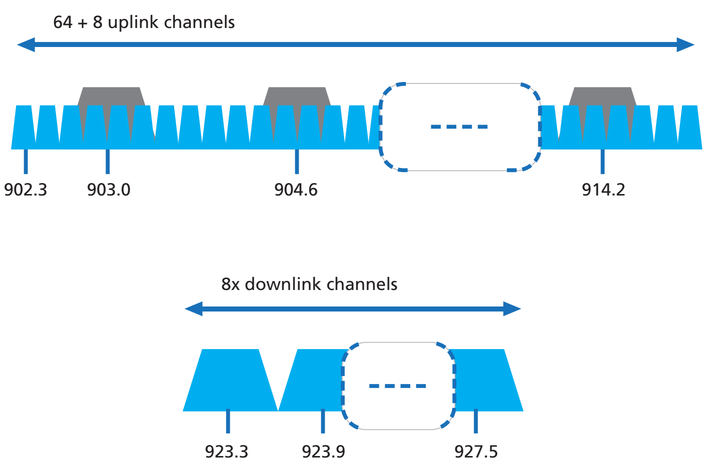

Figure 11: LoRaWAN North American Channel Plan

grade macro gateway and used for outdoor applications only. Cost optimized gateways support eight or 16, 125KHz uplink channels and one or two, 500KHz uplink channels. They typically come in an industrial type version that supports a temperature range of -40C to +85C or an indoor version only that is commonly referred to as a picocell gateway.

Even though the ISM band is considered "License Free," the FCC regulates not only the maximum conducted transmit output power of both end-nodes and gateways, but also the "time-on-air," or dwell time. This is the time that the RF transmitter is actually turned ON. In North America, the maximum dwell time for a 125Khz channel in the ISM band is 400ms (0.4s). Please note that there is no FCC dwell time limitation for the 500Khz channels, so SF11 and SF12 are supported on 500KHz channels.

Table 2 (next page) shows the different data rates and spreading factors that can be used for both uplink and downlink channels. The maximum usable USER payload in a LoRaWAN frame is 242 bytes (represented by the blue payload in Figure 4). As can be seen from Table 2, this maximum user payload is reduced by increasing spreading factor. As discussed earlier in this white paper, an increase in LoRa spreading factor (i.e. from SF7 to SF8) provides an increased sensitivity, and therefore range, but can also increase in the amount of "chips" it sends per bit of user information. Given the fact that the chip

Data Rate (DR)	Spreading Factor	Channel Frequency	Up or Down Link	Bit Rate (bits/sec)	Maximum User Payload Size (Bytes)
0	SF10	125Khz	uplink	980	11
1	SF9	125Khz	uplink	1760	53
2	SF8	125Khz	uplink	3125	125
3	SF7	125Khz	uplink	5470	242
4	SF8	500Khz	uplink	12500	242
5-7					
8	SF12	500Khz	downlink	980	53
9	SF11	500Khz	downlink	1760	129
10	SF10	500Khz	downlink	3900	242
11	SF9	500Khz	downlink	7000	242
12	SF8	500Khz	downlink	12500	242
13	SF8	500Khz	downlink	21900	242

Table 2: Supported Spreading Factors and Data Rates in North America

data rate is constant for a fixed channel bandwidth (BW), in this case 125KHz uplink channel BW, it will take more time for all the chips to be transmitted across the link from the end-node to the gateway. So, as can be seen from the last column in Table 2, the Maximum User Payload is reduced with increasing Spreading Factor in order to prevent the time-on-air not to exceed 0.4 seconds.

It is imperative that end-node sensor designers try to fit their data in the smallest possible payload when using a 125KHz uplink channel, which is 11 bytes at the highest spreading factor, SF10. This does provide the greatest sensitivity, and therefore, longest range. If it is not possible to fit the complete user payload in 11 bytes even after compressing the data, one will have to break up the payload into two or more packets.

IoT TECHNOLOGY STACK

Now that we have completed a high-level overview of the LoRaWAN protocol and network architecture, let us take a step back and look at the IoT end-node technology stack as shown in Figure 12. By definition, in the broadest possible terms, there are five key pieces to any type of IoT solution. First, we have the device hardware and this would be some kind of end-node with either sensors and or actuator capabilities. For example, a temperature sensor for a smart building application or a soil moisture sensor for a smart agriculture application. The device will need to run some kind of software to manage configuration and operation. The end-node will collect the sensor data and then format that data for the right protocol so that it can be transmitted to the Cloud to some kind of user application.

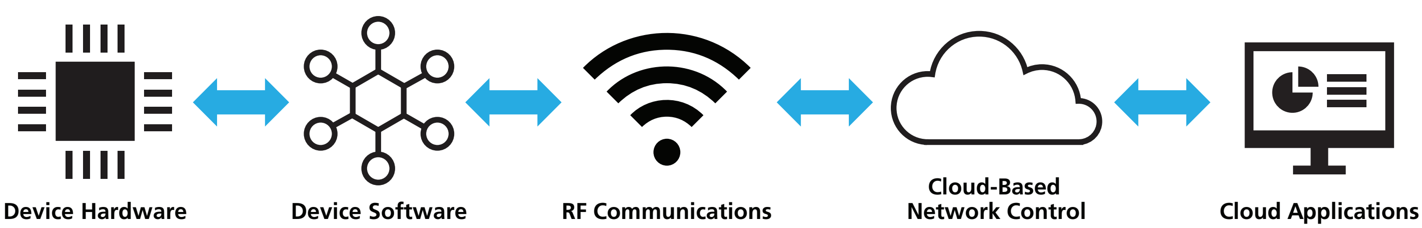

Figure 12: Typical IoT End-Node Technology Stack

Business	LoRa	SigFox	Ingenu (RMPA)	LTE-M/Cat-M	NB-IoT	3G/4G
Module Cost[1]	$4-$6	<$3	$10	$10-$15	$10-$15	$20-$40
Range (miles)	Up to 30 miles	Up to 30 miles	Up to 8 miles	Up to 11 miles	Not commercially deployed	Up to 43 miles
Max Packet Size	242 bytes	12 bytes	6 bytes-10 Kbytes (Flexible)	1000 bytes (Flexible)	Flexible	Flexible
Deployment	Macro-based cable strand-deployable cost optimized pico cells	Macro-based	Macro-based	Macro, small cells	Macro, small cells	Macro, small cells
Channel Diversity	64	360	40 1MHz		N/A	
Frequency	915 MHz (US)					
Battery Life	High	High	Moderate	Low	Low	Low
Standardization	Global standard body through LoRa Alliance	Single Source Network	Single Source Network	3GPP	3GPP	3GPP
Complexity[2]	Low	Low	Moderate	High	High	High
Current US Deployment	Moderate	Low	Low	None	None	High
Global Presence[1]	50	26	29+	143**	None	Saturated

[1]As of 2017 [2]Complexity as defined by how much memory and processing power you need on your end device

Table 3: IoT Technology Comparison

The "RF communications" part in Figure 12 is a crucial piece as it can have major implications on end-node battery life, cost and range. There is no one-size fits all communication technology for IoT. Table 3 compares LoRa, Sigfox, Ingenu, and cellular technologies. Shorter range technologies such as WiFi, BLE, Zigbee, and Z-wave are not included, as they are not well suited for a large scale managed network such as the machine network.

HARDWARE DESIGN CONVERSION

The key objective of any IoT device is to get relatively small amounts of data to a Cloud-based application as described in the previous paragraph. If you are interested in migrating your existing IoT device or end-node to LoRaWAN, you will have to first carefully analyze your message payload requirements as well as the periodicity of these messages. Sending still photographs or even video over WiFi is not something that can be easily implemented using LoRaWAN. If you are able to fit your sensor or other end-node data into an 11 byte payload (or multiple 11-byte payloads) with an average radio message between minutes, hours or days (assuming a battery operated device), converting your existing device to LoRaWAN is possible by following the steps below. Keep in mind that with any battery operated IoT device one of the key goals is to preserve and prolong battery life. This can be achieved by increasing the interval between transmissions of radio messages.

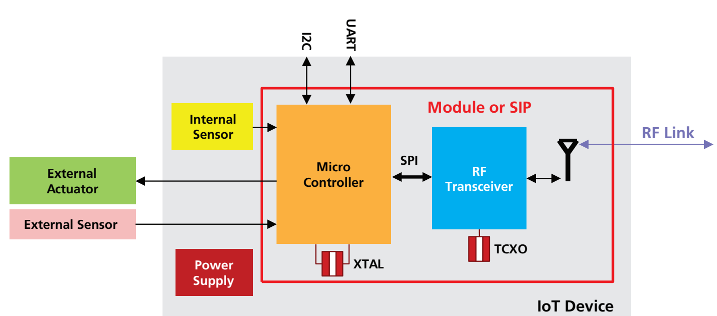

Figure 13: Basic IoT Device Block Diagram

Now that you have concluded that it is possible to migrate your end-device to LoRaWAN, it is time to carefully examine several key aspects of your design. A typical IoT device will have some or all of the following components as depicted in Figure 13:

Power Supply: This can be an internal battery, an external power source or a combination. For example, the device could have a small solar cell connected to a re-chargeable battery. Since one of the key objectives of LPWAN technology is to use sensors that can operate on the same battery for years, it is crucial to estimate the battery life for your particular sensor design. There are five main power consumption modes of the HW that must be determined:

- **OFF/SLEEP:** all electronics are turned off or in some kind of sleep mode.
- **IDLE:** radio, sensor(s), all other electronics turned off except for the microcontroller.
- **RUNNING:** the device is operational (no RF transmissions).
- **LoRa TX:** the device is sending data over the LoRa Radio TX output.
- **LoRa RX:** the device is receiving data over the LoRa Radio RX input.

The next step is to calculate how much current the device will consume in each of these modes. Then, one has to estimate how long the device will be in each of these modes per hour, day or week. One can compare a worst case scenario where all LoRa radio UL packets are sent using the worst case spreading factor (SF10 has the longest TOA (time-on-air), and therefore, uses the most power) versus a scenario where the four spreading factors available for LoRa UL radio messages are used evenly. These two cases will provide you with a worst case battery life and an average battery life.

Semtech offers a LoRaWAN calculator that provides time-on-air data, energy consumption and help with the evaluation of link budgets for their LoRaWAN transceivers:

www.semtech.com/wireless-rf/rf-transceivers/sx1272 (under Products page)

- **Sensor(s):** one or more internal and or external sensors that either have an analog or digital interface. Many microcontroller units (MCUs) these days come with basic analog-to-digital (ADC) functionality, so connecting an analog sensor is straight forward.
- **Actuator(s):** one or more usually external actuators like a lock or a valve.
- **Microcontroller:** runs the software to control the end-node; sampling of the sensor data, formatting of the sensor data into the transmission protocol's payload format, scheduling of radio messages (packets) to some kind of gateway, communicating with the network controller, etc.
- **RF Transceiver (or the radio chip):** this device will convert the digital packetized sensor data to an analog radio signal by modulating it onto an RF carrier frequency. It will also receive incoming RF radio messages, perform de-modulation, convert the analog signals back to digital, and forward the incoming radio messages to the microcontroller.

Keep in mind that often the microcontroller and the RF transceiver are provided as a single "unit", this can either be in a module (2-die packaged in one module) or in a single chip solution. From a HW point of view, the key component that needs to be changed is the RF transceiver. Your current solution could be using BLE, Zigbee, Z-Wave, or even cellular. If a IoT device uses a module type design, i.e., an integrated microcontroller and an RF transceiver, it is relatively straight forward to replace this module with a module from one of the many LoRaWAN module manufacturers such as Laird, Multi-Tech, MicroChip, Murata, or ST Micro.

Using a module instead of putting a discrete transceiver like the Semtech SX1276 directly on your printed circuit board (PCB) will allow you to skip FCC certification as all modules that are already pre-certified. Using a module will also increase your time to market. Once you have found the right module that provides you with the same sensor interfaces as you had on your existing interface you will have to redesign your PCB to fit the new module and or microcontroller and SX1276 combination.

- **Antenna:** this is a key component for reaching the maximum distance in the wireless communication link between the IoT device and the LoRaWAN gateway(s) it will connect to. The goal of an antenna is to transform electrical signals into RF electromagnetic waves,

Antenna Type	Pros	Cons
Wire Antenna	• Extremely cheap • Good performance	• Volume manufacturing performance repeatability
PCB Antenna	• Very low cost if sufficient PCB are available • Decent performance • Relatively small size • Standard design antennas widely available	• Sensitive to quality of ground plane design and placement of nearby components • Requires a relatively large PCB area for a typical IoT PCB Design
Chip Antenna	• Small size • Many different options on the market	• Medium cost and performance • Matching circuitry often required to truly meet published chip antenna specs
Coil (Helical)	• Cheaper than chip antennas	• Medium performance • Care is needed in PCB layout/placement
External (Whip) Antenna	• Best performance • Shorter design cycle	• Highest cost • Placement can be problematic especially on small IoT Devices • Could require a conducted emissions test
IP Antenna (This is a PCB Antenna design that is purchased from a third party)	• Support from IP company • Better performance compared to home grown designs	• High cost compared to standard free PCB antenna designs • Similar cost to Chip antenna

Table 4: Antenna Comparisons

propagating into free space (transmit (TX) mode) and to transform RF electromagnetic waves into electrical signals (receive (RX) mode). Its type, design, orientation, and positioning can make or break your RF link.

- Your current IoT device probably uses one of five types of antennas: a simple quarter wavelength wire antenna, a PCB-antenna that is printed on the circuit board (either home grown or purchased as IP from a third party antenna design company), a chip-based antenna, a simple quarter wave coil antenna, or an external quarter wave whip antenna. Switching to LoRaWAN will require an antenna with a quarter wavelength tuned specifically for the LoRa carrier frequency spectrum of 902-928 MHz. Table 4 shows the pros and cons of each of these types of antennas. Your antenna choice will ultimately become a tradeoff between cost, size and performance.

The easiest and cheapest antenna implementation will be a quarter wavelength wire antenna. One of the issues in IoT end-device volume production will be to minimize variations in antenna placement. Care must be taken in fitting the antenna wire inside the end-node housing to ensure a consistent RF performance on all the devices. This is the cheapest antenna solution available.

A PCB antenna is, in essence, a copper trace on a printed circuit board tuned for the specific carrier frequency range, 902-928 MHz in our case. If there is plenty of PCB space available, this is a viable option for an antenna. Care must be taken in the design as length, width and thickness of the copper trace as all play a part in the efficiency of the antenna. Since the size and shape of the ground plane will affect the antenna radiation pattern, great care must be taken with the ground plane design. The major cost adder of this type of antenna is the additional PCB area needed for the antenna layout. There are companies that sell their PCB antenna's as intellectual property (IP). The cost of these antennas is typically the same as a chip-based antenna. The advantage over a home grown PCB antenna is the available support from the IP antenna design company.

A chip-based antenna takes up a fraction of the board space compared to a PCB trace antenna, but will add bill of material (BOM) and assembly cost. So, if the available board space for the antenna is limited, a chip antenna could be a viable solution. This antenna type allows for small size solutions at 915MHz. The typical cost of a chip antenna is between $0.10 and $0.60. Chip antenna specifications are derived from measurements on test boards with a given thickness, number of layers, and size. Since the PCB size of your IoT end-node design will differ from the size of the test board, care must be taken in the matching circuitry between your radio and the chip antenna or else the published performance parameters will not be met.

A coil (Helical) antenna is a lower cost alternative compared to a chip-based antenna. A helical antenna is simply a length of wire that is wound into a coil. The overall length of the wire determines its resonant frequency, but coiling it can greatly reduce its physical size. Helical antennas have a narrow bandwidth, which is not a problem for LoRaWAN since we only use a narrow part of the ISM spectrum: 903-928MHz. Coil antennas can be easily detuned by the presence of other objects on the PCB, so care must be taken in the layout of the design.

The ideal antenna from a sensitivity and gain perspective is an external quarter wavelength antenna that is tuned for 915MHz. But, this is also the most expensive solution and the one that takes up the most space. An external whip antenna will not only add in the cost of the antenna, but increase the BOM cost since an external connector is required (usually a SubMiniature version A (SMA) connector). Also, it is likely that in order to pass FCC regulations, a conducted emission tests must be performed even if you use a pre-qualified RF module.

Part of the man-hours required for a LoRaWAN end-node conversion project will be taken up by SW development as discussed in the next paragraph.

SOFTWARE DESIGN CONVERSION

A basic IoT device like the one depicted in Figure 13 will have typical software architecture as depicted in Figure 14. First, there will be an assortment of lower level device drivers to connect to USB, a UART, analog or digital Interfaces, in essence providing a HW abstraction layer to the middleware.

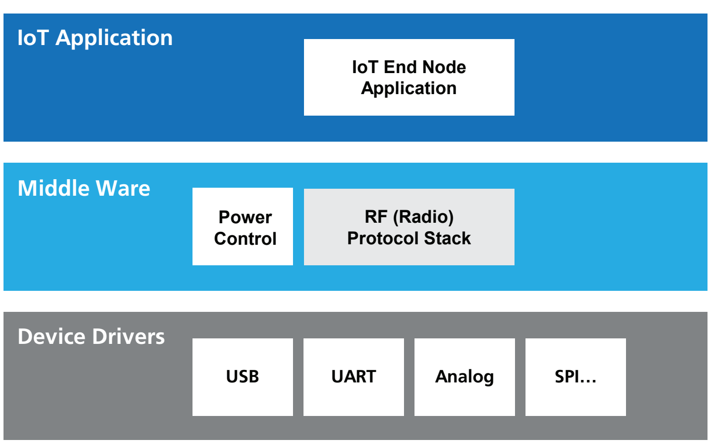

Figure 14: Typical IoT Device Software Architecture

The middleware layer implements any communication protocol type functions. For example, if you have a Bluetooth connected IoT device, the Bluetooth Low Energy (BLE) stack will be implemented in this middleware layer. Finally, the application layer contains the code that implements the device functionality and behavior.

The scope of the SW development will depend on the implementation of the HW architecture of the LoRaWAN end-node. Typically, the SW development in a LoRaWAN device conversion will involve replacing the current communication protocol stack (it could be BLE, ZigBee, Z-Wave, etc.) with a LoRaWAN protocol stack. The good news is that there are HW independent open source code solutions for this stack from either Github or Stackforce:

http://stackforce.github.io/LoRaMac-doc/

https://github.com/Lora-net/LoRaMac-node

A basic LoRaWAN stack takes up about 55-60KB of code space. Care must be taken in selecting the right MCU memory for your design. Since memory is cheap, the more memory the better, especially when considering being able to support SW OTA updates. As mentioned earlier in this white paper, the LoRa Alliance is currently working on an implementation which is loosely based on the sample principle behind the ability to generate all the data on a broken disk drive in a RAID system. An SW OTA update will involve fragmenting a relatively large amount of data in manageable chunks, adding parity information, then broadcasting them to all sensors. Even if certain sensors do not receive all the fragments, they will be able to re-create the missing fragments from the added parity information, just like in a RAID application.

STARTING WITH LoRaWAN

One of the best ways to get started with LoRaWAN is to visit Semtech's LoRa Community:
https://semtech.com/LoRaCommunity

You will find an abundance of information on LoRa, including existing LoRaWAN products such as sensors, gateways, and FAQs.

Another great place to start is the website of the LoRa Alliance:
www.lora-alliance.org

ABOUT SEMTECH

Semtech Corporation is a leading supplier of high performance analog and mixed-signal semiconductors and advanced algorithms for high-end consumer, enterprise computing, communications, and industrial equipment. Products are designed to benefit the engineering community as well as the global community. The Company is dedicated to reducing the impact it, and its products, have on the environment. Internal green programs seek to reduce waste through material and manufacturing control, use of green technology and designing for resource reduction. Publicly traded since 1967, Semtech is listed on the Nasdaq Global Select Market under the symbol SMTC. For more information, visit **www.semtech.com**.

SEMTECH

200 Flynn Road, Camarillo, CA 93012 Phone: (805) 498-2111

The Semtech®, LoRa® and LoRaWAN™ and LoRa Alliance™ logos and marks are trademarks of Semtech Corporation or its subsidiaries.
Semtech reserves the right to make changes to, or discontinue any products described in this document without further notice. Semtech makes no warranty, representation or guarantee, express or implied, regarding the suitability of its products for any particular purpose.
©2018 Semtech Corporation. All rights reserved.