MODULES FOR A WIRELESS WORLD

WIRELESS CONNECTIVITY SOLUTIONS ENCOMPASS A WIDE RANGE OF TECHNOLOGIES, FOCUSED ON HELPING DESIGN ENGINEERS LAUNCH THEIR PRODUCTS ON THE MARKET QUICKLY, AHEAD OF THE GAME

INTRODUCTION

The IoT (Internet of Things) has come to be considered one of the major communication advances in recent years, since it provides the platform for the development of numerous independent interactive services and applications. Industry experts predict that some 500 billion devices will be linked together via the Internet by the year 2020, transforming how people, businesses, and society interact (Ericsson, 2011). Connectivity services and embedded computing, combined with falling prices of communication modules enable a plethora of emerging applications, ranging from utilities and vehicular telematics to healthcare and consumer electronics. A vast amount of activity is ongoing in IoT based product-lines and this activity is expected to grow in years to come, with current projections as high as billions of devices globally, with on average 6-7 devices per person by the end of this decade. IoT technologies such as machine-to-machine communication, complimented with intelligent data analytics, are expected to revolutionize the landscape of a number of industries. Consequently, Wireless technology has taken decisive steps in recent years to ensure sustainable operation in diverse IoT applications. Each device vice functions as a "smart node" in such networks by sensing information and performing low-level signal processing to filter out noise from signals, and to reduce the bandwidth needed for node-to-node communications. In response to this, mobile network operators are beginning to deploy novel IoT solutions that allow their customers to communicate with a centralized "cloud" in a secure manner, to protect, store and process data and to bounce actionable information down to humans.

One very attractive customer scenario is a smart home system where residential gateways have already been used to provide broadband connectivity, quality of service (QoS), and software applications to end users. As these systems continue to evolve, operators can foresee a future where a smart home service gateway will increasingly be employed to control a quantity of additional user function. These include home energy management and automation to social media interaction, connected storage as well as multi-device printing and media streaming. Hence, smart Home Gateway (HG) is rapidly becoming the central hub of the user's home environment, and it is deemed increasingly important to ensure its flexibility and stable and reliable operation. A Wireless Sensor Network (WSN) typically consists of a number of small, inexpensive, locally powered sensor nodes that communicate detected events wirelessly through multi-hop routing. Typically, a sensor node is a tiny device that includes three basic components: a sensing subsystem for data acquisition from the surrounding physical environment, a processing subsystem for local data processing and storage, and a wireless communication subsystem for data transmission. In addition, a power source supplies the energy needed by the device to perform the programmed task. This power source often consists of a battery with a limited energy budget. Therefore, the paramount concern in delivering sustainable home networking is energy efficiency. Indeed, the requirement to achieve continuous wireless connectivity should not, ideally, compromise the need to preserve the battery life of small-scale wireless sensors: conversely, operation time without recharging should be maximized. With this constraint, intricate-to-perform performance trade-offs arise between an individual device and the entire network. Ultimately, the aim is not simply to extend device battery lifetime in order to provide a better user experience, but also to be as efficient as possible with energy as a resource within a broader system and environmental context. Two well-known organizations that manage certification programs nowadays to assure interoperability between wirelessly connected devices are the Wi-Fi Alliance and the *Bluetooth* Special Interest Group (SIG).

WI-FI BASICS

Wired networks differ from wireless ones, which employ radio waves rather than transmitting electrical signals through cables. Wi-Fi, which stands for Wireless Fidelity, is based on the IEEE 802.11 stan-

Two Riverfront Plaza, 7th Floor, Newark, NJ 07102-5490 | na.industrial.panasonic.com | industrial@us.panasonic.com | 1-800-344-2112

dard which was developed as a wireless replacement for the popular wired IEEE 802.3. ethernet standard. As such, it was created from the outset for the purpose of Internet connectivity. Although Wi-Fi technology primarily defines the link layer of local networks, it is nowadays so fully-integrated with the TCP/IP stack, that when people say they are using ‚Wi-Fi' they normally in fact mean that they are using a TCP/IP for internet connectivity. The huge success of Wi-Fi is largely due to the remarkable interoperability programs run by the Wi-Fi Alliance and to the booming market for simple and affordable Internet access. Wi-Fi networks have a star topology, with the AP (Access Point) being the Internet gateway. The out- put power of Wi-Fi is high enough to allow complete in-home coverage in most cases. In businesses and in large buildings, more than one AP is often deployed in different locations inside the building to increase the network coverage. In large concrete buildings, dead spots may exist due to multipath conditions. To overcome such poor signal reception spots, in some cases Wi-Fi products may include two antennas for greater coverage.

Wi-Fi and TCP/IP software is fairly sizeable and complex in scope. For laptops and smartphones with powerful microprocessors (MPUs) and large amounts of memory, this doesn't pose an issue. Until recently, adding Wi-Fi connectivity to devices with low processing power, such as thermostats and home appliances, was neither possible nor economical. Nowadays, silicon devices and modules entering the market embed both the Wi-Fi software and the TCP/IP software inside the device itself. These new products thus eliminate most of the overhead from the MPU and enable wireless Internet connectivity using only a tiny microcontroller (MCU). The continuing integration of these Wi-Fi devices also eliminates the need for radio design expertise, and reduces the barrier of Wi-Fi integration.

Unfortunately, the main downside of current Wi-Fi design is its relatively high power consumption, which in an active data transfer state is of the order of 300 mA, compared to only 3 mA for *Bluetooth*. This is due to its limited range and a more rudimentary design in terms of radio architecture. Even more significant than the power actively transmitted, is the power consumed by a radio network in its idle state. In the case of typical models, most wireless devices only communicate for a small percentage of the time the device is actually on. For some IoT devices which run on batteries and cannot be charged regularly, Wi-Fi can be too power-hungry. Although the peak current on Wi-Fi radios cannot be reduced by much, new silicon devices apply advanced sleep protocols and fast on/off time to reduce the average power consumption dramatically. *Bluetooth* is optimized to run on an extremely low-power state, operating at only 2% power duty-cycle, typically consuming around 50 nA while still remaining available for device discovery and connection setup. In comparison, Wi-Fi is based on CSMA and, although recent implementations support a power saving mode (PSM), the underlying design means that typical power consumption - even though reduced - is still around 1mA in this state. To summarize then, Wi-Fi is the most ubiquitous wireless Internet connectivity technology on the market today. Its high power usage and complexity have been a major hurdle for IoT developers up to now, but new cutting-edge silicon devices and modules are increasingly overcoming these setbacks and enabling Wi-Fi integration into emerging IoT applications and battery-operated devices.

WI-FI ARCHITECTURE

The Open Systems Interconnection Reference Model, or the OSI model, was developed by the International Organization for Standardization. The OSI model is a layered model that describes how information is transferred from an application pro- gram running on one networked computer. In essence, the OSI model prescribes the steps to be taken to transfer data over a transmission medium from one networked device to another and divides the network communications process into seven separate layers (compare Figure 1). From the top down, these layers consist of, respectively: application, presentation, session, transport, network, data link and physical. TCP/IP, or the Internet Protocol suite, underpins the Internet, and provides a simplified concrete implementation of these layers in the OSI model.

Network Access & Physical Layer: This TCP/IP Layer subsumes both OSI layers 1 and 2. The physical (PHY) layer (layer 1 of OSI) pertains to how each device is connected – whether using an optic cable, wires, or radio in the case of wireless network like Wi-Fi (IEEE 802.11 a/b/g/n). At the link layer (Layer 2 of OSI), devices are identifed by a MAC address, and protocols at this level deal with physical addressing, such as how switches deliver frames to devices on the network. By using sequence numbers and acknowledgement messages, TCP can provide a sending node with delivery information about packets transmitted to a destination node. Where data has been lost in transit from source to destination, TCP can retransmit the data until either a timeout condition is reached or until successful delivery has been carried out. TCP can also recognize duplicate messages and will discard them accordingly. If the sending computer is transmitting too fast for the receiving computer, TCP can also communicate deli- very information to the upper-layer protocols and applications it supports. All these characteristics make TCP an end-to-end reliable transport protocol.

Internet Layer: This layer maps to the OSI Layer 3 (network layer), which relates to logical addressing. Protocols at this layer define how routers deliver packets of data between source and destination hosts identified by IP addresses. IPv6 is commonly adopted for IoT device addressing.

Transport: The transport layer (Layer 4 in OSI) is focused on end- to-end communication and provides features pertaining to reliability, congestion avoidance, and guaranteeing that packets will be delivered in the same order that they were sent. The UDP (User Datagram Protocol) is often adopted for IoT transport for performance reasons.

Application Layer: As with the OSI model, the application layer is the topmost layer of TCP/IP model. It combines the application, presentation, and session layer of the OSI model. The Application Layer defines TCP/IP application protocols and how host programs interface with transport layer services to use the network. HTTP/S is an example of an application layer protocol that is widely adopted across the internet.

BLUETOOTH® BASICS

Bluetooth is a type of wireless communication used to transmit audio and data at high speed using radio waves. It is the standard protocol for short range radio communication between many types of devices including mobile phones, computers, entertainment systems and other electronic devices. *Bluetooth* is a replacement for cables between devices, which enables them to communicate with each other within a personal area network. Devices must be within approximately 30 meters of each other.

Bluetooth has a standardized protocol for sending and receiving data via a 2.4GHz wireless link. It consists of a AES-128bit secured protocol, and is perfect for short-range, low-power, low-cost, wireless transmissions between electronic devices. *Bluetooth* networks (commonly referred to as ‚piconets') use a ‚master/slave' model to control when and where devices can send data. Under this model, a single master device can be connected to up to seven different slave devices. Any slave device in the piconet can only be connected to a single master. The master device employs link manager software to identify other *Bluetooth* devices and connects to them with them, enabling data to be sent and received. *Bluetooth* uses spread spectrum frequency hopping technology (SSFH) which operates on multiple frequencies simultaneously to limit interference when using multiple devices.

Bluetooth has been developed over multiple iterations. Table 1 (see next page) displays an overview of the major versions up to the present. Officially, all versions from V2.1 (including Basic Rate) are valid. And although development of the high-speed mode has now been halted, the specifcation still remains valid. Versions up to and including V3 are known as *Bluetooth* Classic and from V4 on as *Bluetooth* Low Energy. *Bluetooth* Low Energy (V4 and later) is optimized for high efficiency requirements as required by devices running for long periods with low current consumption. Devices may support both modes (Dual Mode).

Bluetooth 5 (Low Energy) introduces fundamental improvements, such as a higher data rate. As indicated by the name, the most important feature is the low energy consumption. This is achieved by the device being mostly inactive (99.9 % of the time). Another key objective was a fast connection setup. IoT applications require relatively low data rates. - iVersion 5 features include:

- Twice the data rate, achieved by using a new modulation mode.
- Four times the range thanks to special coding (LE long range).
- Eight times the broadcast capacity through an extension of the advertising procedures (LE advertising extension).
- Higher available transmit power (up to +20 dBm).

OSI MODEL	TCP/IP MODEL	IOT PROTOCOLS
7 Application	Application	HTTPS, XMPP, CoAP MQTT, AMQP
6 Presentation		
5 Session		
4 Transport	Transport	UDP,TCP
3 Network	Internet	IPv6, 6LoWPAN, RPL
2 Data Link	Network access & physical	IEEE 802.15.4 Wifi (802.11 a/b/g/n) Ethernet (802.3) (GSM, CDMA, LTE)
1 Physical		

Figure 1

BLUETOOTH® VERSIONS				
Version	Feature	Discription	Year	Mode
1	Basic Rate	Conventional Bluetooth	1999-2003	Classic
2	Enhanced Data Rate (EDR)	Improved transmission speed	2004-2007	
3	High Speed (HS)	High Speed- mode	2009	
4	Low Energy (LE)	Low energy, IoT, low data rate	2010-2014	Low Energy
5	Improvements for LE	LE: higher data rates, extended range	2016	

*All these features can be guaranteed individually: when combined, performance in terms of range or data must be reevaluated.
*Bluetooth Low Energy is the designation for special extensions starting with Version 4. Versions 4.1 and 4.2 provide minor extensions/impro- vements

Bluetooth Low Energy (BLE) is designed for low energy applications. The BLE is particularly handy in day-to- day life and is thus commonly used on mobile devices, like smartphones and tablets, where the expertise of the software developers is focused on the application level (Android/Java, iOS/Objective-C, Windows Phone/C#, ...). From this viewpoint the BLE is quite similar to many other event-oriented solutions. *Bluetooth* Low Energy significantly reduces the power consumption of *Bluetooth* devices and enables years of operation by the use of coin cell batteries. Supported by the new generation of smart- phones and tablets (compare Figure 2), *Bluetooth* Low Energy has accelerated *Bluetooth* market growth and enabled a huge range of new applications spanning health and fitness, toys, automotive and industrial spaces. *Bluetooth* low energy has also introduced new advances in proximity technology that open the door to location-based services such as beaconing and geo-fencing. *Bluetooth* Low Energy supports an unlimited number of devices but the practical number of simultaneously connected devices is between 10 and 20.

Bluetooth Classic refers to any *Bluetooth* device that does not utilize the BLE link layer. This classification encompasses devices supporting *Bluetooth* versions 1.0 to 2.1, including Enhanced Data Rate (EDR) mode. EDR offers short-range wireless transmissions of a data rate up to 2 Mbps net. At these rates, *Bluetooth* supports applications such as audio-streaming, computer networking, or large file transfers in a Personal Area Network (PAN). Devices using *Bluetooth* Classic are identified by their 6-byte *Bluetooth* Device Address (BD ADDR) and perform up to 1,600 frequency hops every second across 79 channels.

The hopping scheme and clock are negotiated during the connection process and controlled for the duration of the connection by the requesting device. BTC can support up to eight devices connected in a star network simultaneously. One of the advantages of the *Bluetooth* standard is that it includes application profiles. These profiles define in great detail how applications exchange information to achieve specific tasks. To name one example, the Audio/Video Remote Control Profile (AVRCP) defines how a *Bluetooth* remote control interfaces with audio and video equipment to relay commands like play, pause, stop, etc. The comprehensive certification programs defined by the *Bluetooth* SIG cover the entire protocol stack as well as the application profile, helping *Bluetooth* achieve excellent interoperability in the market.

There are essentially two strains of *Bluetooth*: the older legacy or 'classic' strain that encompasses versions 1.0 through 3.0 (including EDR), and the low-energy *Bluetooth* that includes versions 4.0, 4.1, and 4.2. The low-energy version uses a different radio technology to the classic strain. It employs frequency-hopping spread spectrum (FHSS) over the 2.4 to 2.483 GHz spectrum, but uses 40 2 MHz-wide channels rather than the 79 1 MHz channels of classic *Bluetooth*. Maximum data rate is 250 kb/s. Most of the newer *Bluetooth* chips actually contain both types of radios.

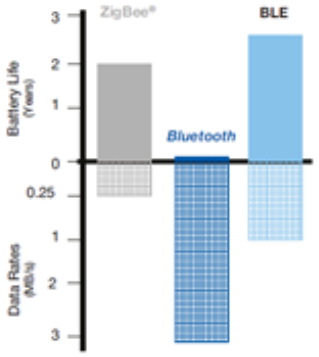


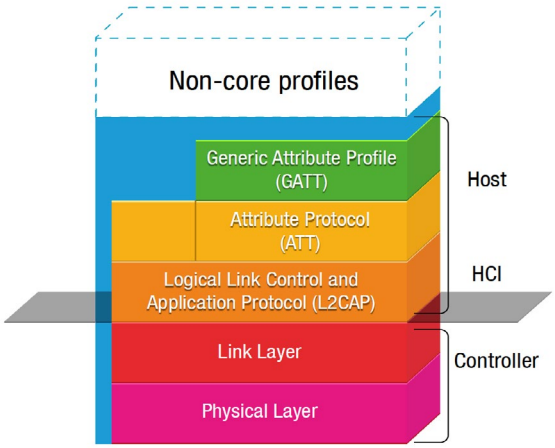
Figure 2

BLUETOOTH® ARCHITECTURE

Bluetooth permits devices to establish ad hoc networks. Ad hoc networks allow the establishment of an easy connection between devices in the same physical area (e.g. the same room) without the use of any infrastructure devices. A *Bluetooth* client is simply a device with a *Bluetooth* radio and software incorporating the *Bluetooth* protocol stack and interfaces. As with classic *Bluetooth*, the BLE protocol stack is comprised of two main components: the Controller and the Host. The Host and Controller relay information to each other using standardized communication over the Host Controller Interface (HCI). This standardized HCI allows host and controller from different product manufacturers to interoperate. In some cases, the host and controller functions are integrated into a single device.

The Controller comprises the Physical Layer and the Link Layer, and is typically implemented as a small System-on- Chip (SOC) with an integrated radio. The Host runs on an application processor and includes upper layer functionality, i.e., the Logical Link Control and Adaptation Protocol (L2CAP), the Attribute Protocol (ATT), the Generic Attribute Profile (GATT), the Security Manager Protocol (SMP) and the Generic Access Profile (GAP). Communication between the Host and the Controller is standardized as the Host Controller Interface (HCI). Finally, non-core profiles (i.e., application layer functionality not defined by the *Bluetooth* specification) can be used on top of the Host. Figure 3 illustrates the BLE protocol stack. Although some of the BLE Controller features are inherited from the classic *Bluetooth* Controller, both types of Controller are currently incompatible. Hence, a device that only implements BLE (which is referred to as a single- mode device) cannot communicate with a device that only implements *Bluetooth* Classics. It is foreseeable that many devices will eventually implement both the *Bluetooth* Classic and the BLE protocol stacks. These devices are called dual-mode devices.

Physical Layer: BLE's physical layer (PHY) contains the analog communications circuitry responsible for translation of digital symbols over the air. It is the lowest layer of the protocol stack, and provides its services to the Link Layer. BLE operates in the 2.4 GHz Industrial Scientific Medical (ISM) band and defines 40 Radio Frequency (RF) channels with 2 MHz channel spacing. There are two types of BLE RF channels: advertising channels and data channels. Advertising channels are used for device discovery, connection establishment and broadcast transmission, whereas data channels are used for bidirectional communication between connected devices. Although every application is different, advertising the most important or unique services provided by the peripheral is the easiest way to connect to it, and it makes sense in a lot of products. When an iPhone or Android is searching for other devices, it can use the custom service UUID to find the exact devices it wants to



GAP: Generic Access Profile
SMP: Security Manager Protocol
HCI: Host Controller Interface

Figure 3

talk to and filter out other devices. Three channels are defined as advertising channels. These channels have been assigned center frequencies that minimize overlapping with IEEE 802.11 channels 1, 6 and 11, which are commonly used in several countries. An adaptive frequency hopping mechanism is used on top of the data channels to deal with interference and wireless propagation issues, such as fading and multipath. This mechanism selects one of the 37 available data channels for communication during a given time interval. The three advertising channels are spread out across the BLE spectrum to minimize interference potential from other services. A BLE device advertises on an advertising channel for a fixed time period and then invokes a random delay before switching to the next advertising channel. Since battery life is a prime concern for BLE devices, adding more advertising channels could be problematic as advertising would then reduce battery life. More advertising channels would also mean fewer available data channels. Whilst the use of fewer advertising channels would increase battery life, it would also result in more advertising collisions between devices in near proximity. This would increase the latency of device connections. All physical channels use a Gaussian Frequency Shift Keying (GFSK) modulation, which is simple to implement. The modulation index is in the range between 0.45 and 0.55, which allows reduced peak power consumption. The physical layer data rate is 1 Mbps.

Link Layer: The BLE link layer is the part that directly interfaces to the PHY. It is responsible for advertising, scanning, and creating/maintaining connections. In BLE, when a device only needs to broadcast data, it transmits the data in advertising packets through the advertising channels. Any device that transmits advertising packets is termed 'an advertiser'. The transmission of packets through the advertising

channels takes place in intervals of time called 'advertising events'. Within an advertising event, the advertiser uses each advertising channel sequentially for packet transmission. Devices that only aim at receiving data through the advertising channels are called 'scanners'. Bidirectional data communication between two devices requires them to connect to each other. The creation of a connection between two devices is an asymmetric procedure by which an advertiser announces through the advertising channels that it is a connectable device, while the other device (referred to as an 'initiator') listens for such advertisements. When an initiator finds an advertiser, it may transmit a Connection Request message to the advertiser, which creates a point-to-point connection between the two devices. Both devices can then communicate by using the physical data channels. The packets for this connection will be identified by a randomly generated 32-bit access code.

BLE defines two device roles at the Link Layer for a created connection: the 'master' and the 'slave'. These are the devices that act as initiator and advertiser during the connection creation, respectively. A master can manage multiple simultaneous connections with different slaves, whereas each slave can only be connected to one master. Thus, the network composed by a master and its slaves, which is called a piconet, follows a star topology. Currently, a BLE device can only belong to one piconet. In order to save energy, slaves rest in sleep mode by default and wake up periodically to listen for possible packet receptions from the master. The master determines the instants in which slaves are required to listen, and thus coordinates the medium access by using a Time Division Multiple Access (TDMA) scheme. The master also provides the slave with the information needed for the frequency hopping algorithm (including the map of data channels to be used) and for the connection supervision. The parameters related to the management of a connection are transmitted in the Connection Request message and can be updated during the connection for various reasons (e.g. using a new data channel map due to a change of the interference pattern).

Once a connection between a master and a slave is established, the physical channel is divided into non-overlapping time units called 'connection events'. Within a connection event, all packets are transmitted using the same data channel frequency. Each connection event begins with the transmission of a packet by the master. If the slave receives a packet, the slave must send a packet to the master in response. However, the master is not required to send a packet upon receipt of a packet from the slave. As a minimum, an Inter Frame Space (IFS) of 150 µs must pass between the end of the transmission of a packet and the start of the next one. While master and slave continue to alternate in sending packets, the connection event is considered to be open. Data channel packets include a More Data (MD) bit which signals whether the sender has more information to transmit. If none of the devices has any more data to transmit, the connection event will be closed and the slave will not be required to listen until the beginning of the next connection event. Other circumstances that force the end of a connection event include the reception of two consecutive packets with bit errors by either the master or the slave, as well as the corruption of the access address field of a packet sent by any device. In order to allow bit error detection, all data units include a 24-bit Cyclic Redundancy Check (CRC) code.

In the case of a new connection event, master and slave adopt a new data channel frequency, which is computed using the frequency hopping algorithm. The interval between the start of two consecutive connection events is specified by the connInterval parameter, which is a multiple of 1.25 ms in the range between 7.5 ms and 4 s. Another important parameter is connSlaveLatency, which defines the number of consecutive connection events during which the slave is not required to listen to the master and thus can keep the radio turned off. This parameter is an integer between 0 and 499 and should not cause a supervision timeout. A supervision timeout happens when the time since the last received packet exceeds the connSupervisionTimeout parameter, which is in the range between 100 ms and 32 s. The purpose of this mechanism is to detect the loss of a connection caused by severe interference or the movement of a device outside the range of its peer.

Link Layer connections use a stop-and-wait flow control mechanism based on cumulative mutual responses, which at the same time provides error recovery capabilities. Each data channel packet header contains two one-bit fields called the Sequence Number (SN) and the Next Expected Sequence Number (NESN). The SN bit identifies the packet, whereas the NESN indicates which packet from the peer device should be received next. If a device successfully receives a data channel packet, the NESN of its next packet will be gradually increased, and that packet will serve as an receipt of delivery. Otherwise, if a device receives a packet with an invalid CRC check, the NESN of the received packet cannot be relied upon. This forces the receiving device to resend its last transmitted packet, which serves as a negative acknowledgment.

L2CAP: (Logical Link Control and Adaption Protocol): L2CAP acts as a protocol multiplexer and handles segmentation and reassembly of packets. It also provides logical channels, which are multiplexed over one or more logical links. The L2CAP used in BLE is an optimized and simplified protocol based on the classic *Bluetooth* L2CAP. Typically, application developers do not need to worry about the details of

interacting with L2CAP layer. The interactions are handled by the *Bluetooth* stack, and the details of the L2CAP operation are not covered in this paper.

ATT (Attribute Protocol): The attribute protocol provides the means of transmitting data between *Bluetooth* devices. It relies on a *Bluetooth* connection and provides procedures to read, write, indicate and notify attribute values over that connection. The client or server role is determined by the GATT, and is independent of the slave or master role. The client can access the server's attributes by sending requests, which trigger response messages from the server. For greater efficiency, a server can also send to a client two types of unsolicited messages that contain attributes: (i) notifications, which are unconfirmed; and (ii) indications, which require the client to send a confirmation. A client may also send commands to the server in order to write attribute values. Request/response and indication/confirmation transactions follow a stop-and-wait scheme.

GATT (Generic Attribute Profile): GATT is an acronym for the Generic Attribute Profile, and it defines the way that two *Bluetooth* Low Energy devices transfer data back and forth using concepts called 'Services and Characteristics'. It makes use of the Attribute Protocol (ATT), which is used to store Services, Characteristics and related data in a simple lookup table using 16-bit IDs for each entry in the table. The most important thing to bear in mind with GATT and connections is that connections are exclusive. In other words, a BLE peripheral can only be connected to one central device (a mobile phone, etc.) at a time!

GAP and Application Profiles (Generic Access Profile): The GAP layer provides platform for low energy *Bluetooth* devices to advertise. At the highest level of the core BLE stack, the GAP specifies device roles, modes and procedures for the discovery of devices and services, the management of connection establishment and security. The BLE themselves, or other devices, make device discovery, open and manage connections and broadcast data, GAP defines four roles with specific requirements on the underlying controller: Broadcaster, Observer, Peripheral and Central. A device in the Broadcaster role only broadcasts data (via the advertising channels) and does not support connections with other devices. The Observer role is complementary for the Broadcaster, i.e., its purpose is to receive the data transmitted by the Broadcaster. The Central role is designed for a device that is in charge of initiating and managing multiple connections, whereas the Peripheral role is designed for simple devices which use a single connection with a device in the Central role. Therefore, the Central and Peripheral roles require the device's controller to support the master and slave roles, respectively. A device may support various roles, but only one role can be adopted at any given time. Finally, since certain types of applications may benefit from reusing common functionality, additional profiles can be built in addition to the GAP. *Bluetooth* follows a profile hierarchy, whereby a new profile including all the requirements of an existing profile can be defined. A highest-level profile that specifies how applications can interoperate is called an 'application profile'. Application profiles, which are also specified by the *Bluetooth* SIG, favor interoperability between devices from different manufacturers.

Security: BLE provides various security services for protecting the information exchange between two connected devices. Most of the supported security services can be expressed in terms of mutually-exclusive security modes called LE Security Mode 1 and LE Security Mode 2. These two modes provide security functionality at the Link Layer and at the ATT layer, respectively. The BLE Link Layer supports encryption and authentication by using the Cipher Block Chaining Message Authentication Code (CCM) algorithm and a 128-bit AES block cipher. When encryption and authentication are used in a connection, a 4-byte Message Integrity Check (MIC) is appended to the payload of the data channel PDU. Encryption is then applied to the PDU payload and MIC fields. It is also possible to transmit authenticated data over an unencrypted Link Layer connection. The signature is computed by applying an algorithm that uses 128-bit AES as the block cipher. One input to the algorithm is a counter which is used in order to provide protection against replay attacks. If the receiver verifies the signature, it assumes that the data have been sent by a trusted source. In addition to the described services, BLE supports a mechanism called 'privacy feature' which allows a device to use private addresses and frequently change them. The privacy feature mitigates the threat by which an adversary can track a BLE device. The private addresses are generated by encrypting the public address of the device which can be unlocked by a trusted device that has been provided with corresponding encryption key.

Each security mode corresponds to different levels. These express requirements as to the type of pairing that has to be used. Pairing is a procedure by which the devices generate and distribute key material. Pairing comprises three phases. In the first phase, the two connected devices announce their input/output capabilities and, based on these, they choose a suitable method for the second phase.

The second phase aims to generate the Short-Term Key (STK), which will be used in the third phase to secure the distribution of key material. In the second phase, the pairing devices first agree on a Temporary Key (TK), by means of the Out Of Band, the Passkey Entry or the Just Works methods. The Out of Band method uses out of band communication means for the TK agreement. In the Passkey Entry method, the user enters six numeric digits as the TK between the devices. When none of the first two methods can be used, the Just Works method is employed, although it is not authenticated and it does not provide

protection against Man In The Middle (MITM) attacks. Based on the TK and on random values generated by each pairing device, the STK is obtained by both devices, which leads to the end of the second phase.

In the third phase, each end-point of the connection may distribute to the other endpoint up to three 128-bit keys called the Long-Term Key (LTK), the Connection Signature Resolving Key (CSRK) and the Identity Resolving Key (IRK). The LTK is used to generate the 128-bit key employed for Link Layer encryption and authentication. The CSRK is used for the data signing performed at the ATT layer. The third key (i.e., the IRK), is used to generate a private address on the basis of a device public address. The message exchange required for distributing the LTK, the CSRK or the IRK is encrypted by using the STK obtained in the second phase. The Security Manager Protocol (SMP) carries out the message exchange of the three described pairing phases. SMP operates on top of a fixed L2CAP channel.

A weakness that currently exists in BLE is the fact that none of the pairing methods is protected against passive eavesdropping. As such, an adversary who obtains the pairing messages can then determine the LTK, the CSRK or the IRK. In version 4.0 and 4.1 of the core specification, *Bluetooth* with its lower energy functionality uses the Secure Simple Pairing model (referred to as LE Legacy after *Bluetooth* 4.2 release), in which devices choose one method from Just Works, Passkey Entry and OOB based on the input/output capability of the device. With the release of the *Bluetooth* Core Specification version 4.2, security has been greatly strengthened by the new LE Secure Connections pairing model. In this new model, the numeric comparison method is added to the other three methods and the Elliptical Vurve Hellman-Diffie (ECDH) algorithm is introduced for key exchange in this process. For a consumer using LE legacy pairing, each of these association models is similar to BR/EDR Secure Simple Pairing with the notable exception that neither Just Works nor Passkey Entry provide any protection against passive eavesdropping. In LE Secure Connections pairing, the four association models are functionally equivalent to BR/EDR Secure Connections. The use of each association model is based on the I/O capabilities of the device.

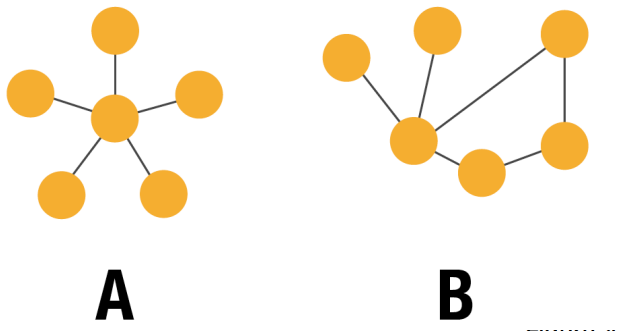


Figure 4

BLUETOOTH® STAR-TYPE TOPOLOGY VS. MESH

Bluetooth Classic is a star-type topology (Figure 4A) in which all devices are connected to a central hub. As the devices cannot themselves act as hubs, the only way to extend the network is to connect additional devices to the central hub. Whilst this can be accomplished in a wired network (although requiring lots of cable), the range of a wireless star-type network is limited, so its maximum distance is determined by the furthest connected device. A much better solution is the mesh network (Figure 4B) as it has been implemented starting with BLE in which all devices communicate with each other. This makes the size and area covered by the network almost unlimited.

There are two types of mesh networks. The first is a routed mesh, where individual devices in the network have paths of conversation. The second one is a flooding mesh, where every device on the network can broadcast its messages to every other device. Some of the original *Bluetooth* mesh efforts behaved this way, which led to challenges where too many devices where added to the network. It is also less power efficient. The *Bluetooth* SIG has split the difference, deciding on a managed flood for its network. This means that certain devices can pass along messages but not every device can do so. In many cases, those devices capable of passing along a message will be wired into a power source, such as a light bulb. BLE has a mechanism for devices to broadcast information relating what they are exactly and their capabilities so that other devices can synchronize with them. Until now, those messages could only contain 31 bytes of information. BT5 messages can be up to eight times that size, making it easier to share information concerning the location and condition of business assets, such as medical devices in hospitals. This mesh software is designed to operate on any *Bluetooth* Low Energy radios that are 4.0 or later. The original audio- streaming *Bluetooth* radios need not apply. Many chip vendors have anticipated the needs of *Bluetooth* mesh and have been building chips that can be updated over the air. This means that before long, products will receive an update that changes them from *Bluetooth*'s original flooded routing approach to the new, managed flood approach. In practical terms, your *Bluetooth* mesh will be limited to a specific building or location. *Bluetooth* 5 is designed to create world class IoT functionality for multiple classes of devices.

DESIGNING WITH WIRELESS MODULES OR WIRELESS SOC'S

When adding wireless functionality to products, one of the first decisions that has to be made is whether to use a system-on-a-chip (SoC) or a wireless module. A SoC consists of a single chip where the vendor has packaged all features one will usually require to address a specific market. There are several *Bluetooth* Low Energy SoCs in the market which integrate in a single chip, or more microcontrollers the following features; static RAM memory; Flash memory; *Bluetooth* controller; RF transceiver; voltage regulators and many extra peripherals (analog to digital converters, timers, cryptographic processors, etc.) A BLE module is either available with a fully-contained BLE transceiver plus embedded controller and built-in antenna that is preprogrammed to handle all a design's radio interactions. Or as a BLE module that serve purely as an IO device for a host controller (HCI module), making the BLE connection the logical equivalent of a serial port for design purposes. Other modules are also able to operate in a stand-alone (hostless) manner and make available their processor and other IO resources to developers to run application code. Both module types come pre-certified with both the *Bluetooth* SIG (for interoperability) and various regulatory agencies. Sometimes a single-chip solution is not always possible or the preferred option. Many semiconductor companies have released highly integrated devices (modules) to support new IoT products. For example, new low-power System-on-Chip (SoC) integrated module support both wireless protocols, and external sensing and communication technologies. The inherent complexity of RF design, wireless connectivity and qualifying compliances are easily overcome by using these integrated modules. Or to put it in a nutshell:

Modules drive Rapid time-to-market, Wide range of products while ICs move with "less expensive, custom design products.

In most cases, an SoC-based design can be cost optimized, size optimized and configured to the specific needs of the end product. Conversely, a module approach is likely to be easier to implement, will be pre-certified and require less resources. In the end, only the individual consumer may effectively judge which approach best meets their particular needs. However, here are a few incremental costs to be taken into account:

1. RF Design. Designing your own radio? This is a required skill for an SoC design. Good RF engineers are in high demand and can be costly to hire. Modules are self-contained, typically with an antenna, so there is much less engineering involved. Just plug-and-play Panasonic's SMT module for RF/ wireless functionality right off the bat.

2. Lab Equipment. Extensive RF analysis which requires very specialized equipment such as spectrum analyzers, anechoic chambers, etc. The cost of purchasing such equipment is relatively high, but renting it can also be prohibitive. Modules manufacturers have carried out much of this work.

3. PCB Layout and Antenna Design. Although SoC manufacturers typically include detailed reference designs and layout guidelines, RF developments rarely go precisely according to plan. A few quick-turn PCB test runs can boost costs considerably.

4. Regulation and Certification. Wireless products must pass stringent emissions tests and the standards of these can vary globally. Certification testing is also required to ensure interoperability. All of these regulatory measures have their associated costs. Typically, Modules will already have been through this process and thus have their test results and certification readily available for inspection.

5. Time to Market. This is a key factor. The less wireless experience one has, the more likely it is that market windows of opportunity can be missed. Chip solutions require individually-tested end-products while a module is already fully tested.

6. Supply Management and Assurance. A module is available to purchase as a single unit as opposed to SoC-based designs, which require external components. For low-volume production runs, modules can mitigate supply risk. Sourcing a single module is much simpler than sourcing all the individual components to put on an SoC on the board.

Every project is different. So therefore, are the costs, which can obviously vary immensely depending on the available expertise and equipment, and its application. The chip-based design cost, for example, is suitable for a company building up a team with the requisite expertise. Subsequent design should work out to be significantly cheaper. The costs also assume that each design will be right the first time round and not have to go through multiple spins or product tests. Chip-based designs carry a greater risk of re-spins, especially if the design is a company's first attempt at RF. In practice, designs with total production volumes of fewer than 50,000 units will generally benefit from using a module. Above 200,000 units, it is probable that a chip-based design will be more economical (however, it is of course not only a question of quantity but also know-how) - as exemplified in Figure 5.

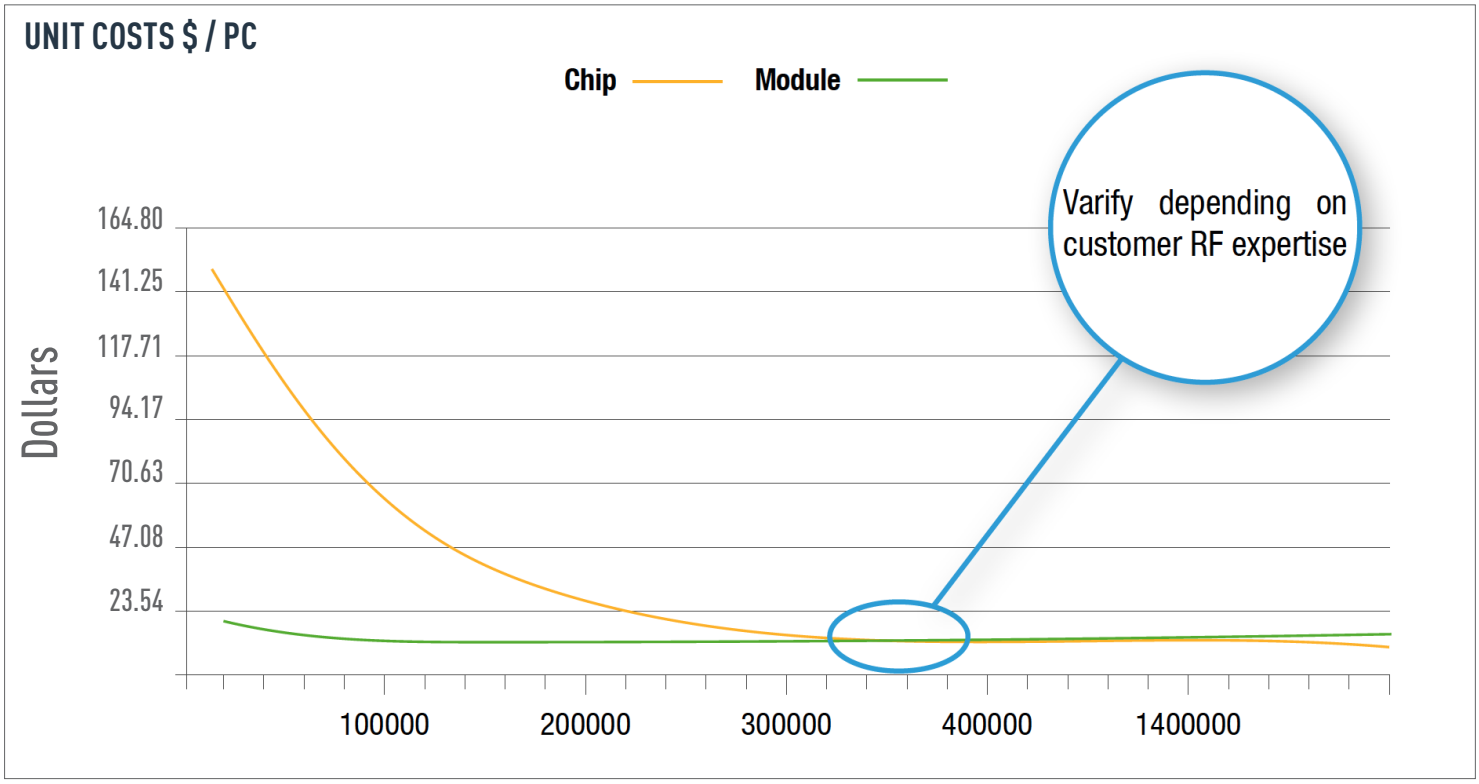


Figure 5

With a chip solution additional costs occur, which are not considered in above calculation:

- Delayed market entrance because of longer development results in loss of market share
- Opportunity costs: tied up rescources and capital
- Higher risk due to technical not core competency area

With the increasing quantity of wireless nodes, a modular approach is becoming increasingly popular and now represents a growing proportion of today's market. As of now, modules represent 32% of the overall low power wireless market, according to IHS. In order to meet varying customer expectations and needs, Panasonic compliments its' well-established range of ICs with Wireless modules, providing an easy solution or an intermediate step before chip-down implementation.

OpenDOF – YOUR ECOSYSTEM FOR THE INTERNET OF THINGS

Panasonic continues to contribute positively to the 'Internet of Things' ecosystem by making its solutions available for use without license fees. Specifically, Panasonic provides a peer-to-peer IoT framework for machine-to-machine communication, upon which other companies can build. Panasonic also provides the source code to the OpenDOF Project, a non-profit organization that it established for this purpose. OpenDOF (Open Distributed Object Framework) provides the missing link for IoT applications: it is a free, open-source framework, which connects different products based on different networking technologies and permits the exchange of data. The OpenDOF Project's framework brings the following core capabilities to developers:

Security: For the OpenDOF Project, security is not merely an afterthought; it was constructed from the beginning with security in at the forefront. The OpenDOF Project supports a wide range of encryption algorithms up to 256 bits in strength. The OpenDOF Project also builds in the authentication and access control of components in the service. The OpenDOF Project's framework ensures that developers integrate these capabilities from the beginning of development, making them simple to enable.

Distributed Capabilities: The OpenDOF Project does not rely upon a centralized cloud-controlled infrastructure. While the cloud is likely to be an integral part of many services using the OpenDOF Project, services could also be implemented where components communicate with each other directly in a peer-to-peer fashion. This could lead to enhanced security since there would be no centralized location for rogue actors to attack. Components in a network that has been enabled by the OpenDOF Project can be meshed together so messages can be relayed between a number of components. This allows services to use broadly distributed components, even when globally distributed. From a security and survivability perspective, the way OpenDOF Project supports distributed topologies and minimizes dependence on a centralized infrastructure.

Interoperability: The OpenDOF Project framework ensures a consistent approach to the descriptions that service components use to communicate with each other. This means that services using the OpenDOF Project can avoid using translators to change descriptions between components that otherwise could not communicate with each other. At its core, the OpenDOF Project framework is a complete protocol stack which can be used equally among the components that are typically used in device-focused connected services: clients for user interaction; gateways that work with multiple devices; and cloud services that may be used to gather, analyze, and propagate data from devices.

CONCLUSION

Bluetooth and Wi-Fi are both platforms that provide Wireless communication, but the difference between the two mainly stems from what they are designed to do and how they are used. The key difference is that *Bluetooth* is primarily used to connect devices without the use of cables, while Wi-Fi provides high-speed access to the internet. Due to the short range and low power consumption of *Bluetooth*, it is best suited to formats such as portable devices, smart health, body sensor network (BSN), smart vehicle application etc. Wi-Fi is best suited for standalones (mostly in smart home applications) and mobile devices because it can implement TCP/IP and therefore the devices or nodes can connect to the Internet directly. Thus, both the functionality and nature of use determines whether *Bluetooth* or Wi-Fi will be the best solution in a given context.

Part Number	ENW49C01A3KF (FCC) ENW49C02A3KF (CE)	ENWF9202A1EF (CE) ENWF9201A1EF (FCC)
RF Category	Wi-Fi Embedded 802.11 b/g/n	Wi-Fi Combo Dual Band 802.11 a/b/g/n (2.4 GHz & 5.0 GHz) + *Bluetooth*® Dual Band BLE v4.2 class 1
Software / Profile	Full Embedded	Linux
Used ICs	88MW300	88W8977
Size [mm]	29.0 x 13.5 x 2.66	17.5 x 10.0 x 2.6
Rx Sensitivity [dBm]	-97 @ 1M-DSSS	-98 @ 1M-DSSS
Tx Power (max.) [dBm]	+16 @ 11b	+17 @ 11b
Power Supply [V]	3.0 to 3.6	1.8 to 3.3
Current Consumption (max.)	Tx: 310mA @ 11Mbps Rx: 75mA @ 11Mbps Power Down: < 1 mA	Tx: 400mA @ 11Mbps Rx: 70mA @ 11Mbps Power Down: tbd
Interfaces	2 x UART	SDIO 3.0, HS UART
Microcontroller and Memory	Cortex M4F, 4 MB Flash	
Operating Temp. [°C]	-40 to +85	-30 to +85
Evaluation Kit	ENW49C01AYKF (EMK)	ENWF9201AZEF (ETU) ENWF9201AXEF (KIT)