Appendix F – Theory of Operation

1. Components

The BlueMeter Hub is composed of two major components: an embedded computer and a wireless module. The embedded computer is configured with Eka Systems proprietary software, so that a user may program the computer to perform tasks automatically or interactively. The wireless module uses a Bluetooth transceiver and interfaces to the embedded computer through an RS-232 interface. Using the Bluetooth transceiver, the embedded computer communicates wirelessly with other Eka Systems devices.

The wireless module is permanently attached to a serial port on the embedded computer. Besides the Bluetooth transceiver, the wireless module contains line-driving components to interface the serial port with the Bluetooth transceiver. The Bluetooth transceiver has a maximum transmit power of 20 dBm and is compliant with the Bluetooth Specifications, Version 1.1.

2. Operations

The BlueMeter Hub communicates and collects data from up to 64 electricity meters. At start up the radio receiver on the Hub is activated. The Hub initially discovers wireless meters within its range. A wireless network administrator may assign a meter to a BlueMeter Hub by designating a *managed* status to the Hub. For each managed meter, Hub collects data transmitted by the meter. The Hub also calculates and stores interval data. A proprietary application software allows configuring the Hub to detect and report network status alarms, power failures, communications failures and low battery alarms. The application software also maintains logs and displays network alarms and network events.

Another application software communicates with the Hub and provides energy management information to the end-user. For example, the information includes, energy usage, peak demand, load duration and power quality. This information provides insight to various aspects of a utility bills including, energy charges, demand charges, and peak usage charges. The connection may be over a serial cable or over Ethernet. The user may select a meter managed by the Hub and request an on-demand meter read. For interval data collection, the end-user identifies frequency of interval data collection for each of the desired meter. Once the interval collection is defined for a meter, the application software collects interval data at those intervals. The collected data may be listed in a tabular or graphical format. Figure 1 shows the setup of the BlueMeter Hub.

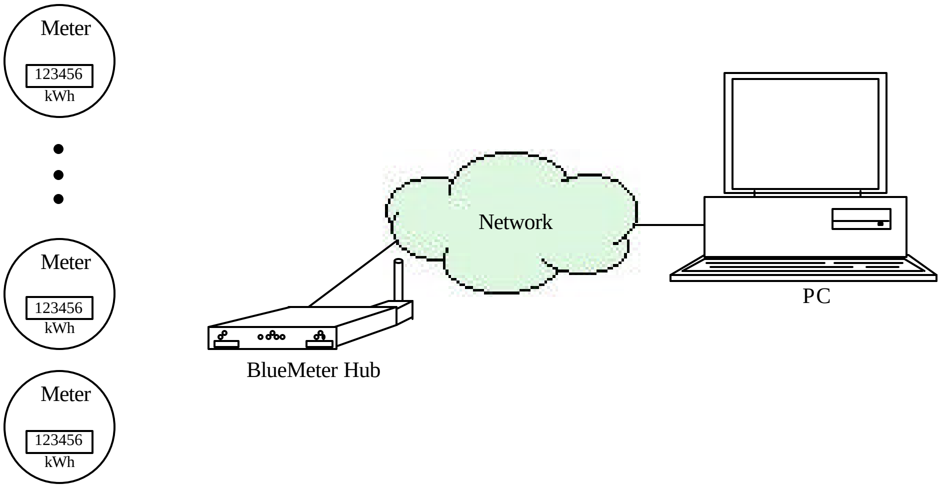
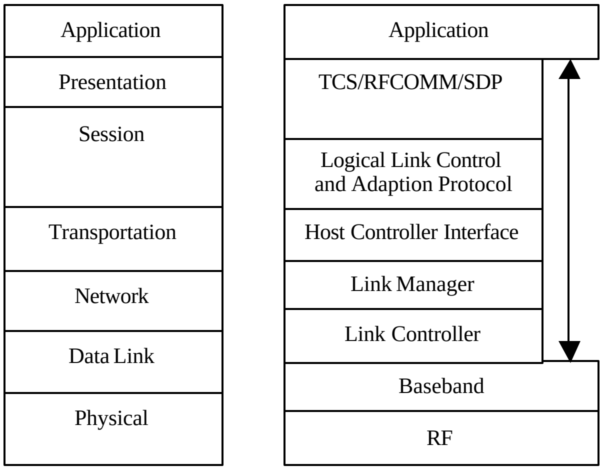


Figure 1: Setup of the BlueMeter Hub

3. Radio Functions

The BlueMeter Hub uses a Bluetooth transceiver for data transportation over the air. Bluetooth is a set of wireless specifications that can be mapped to the ISO Open System Interconnection (OSI) Reference Model in the manner shown in Figure 2.



OSI Reference Model

Partial Bluetooth protocol stack

Figure 2: OSI reference model and the full Bluetooth protocol stack

It is also possible to use only a partial set of the full Bluetooth functionalities in the manner shown in Figure 3; the BlueMeter Hub uses the Bluetooth transceiver only for data transportation and falls into this category.

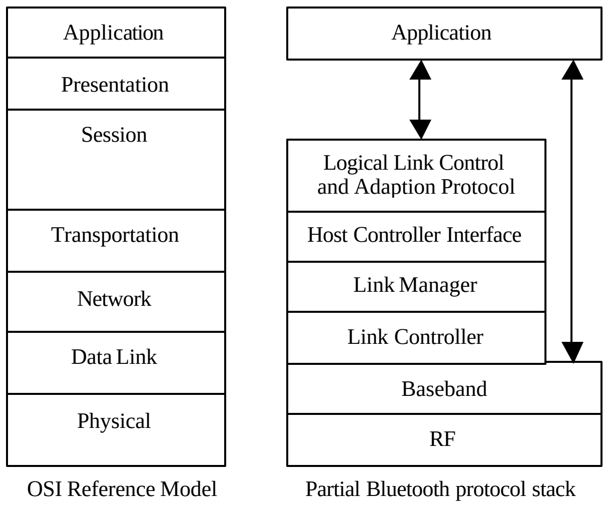


Figure 3: OSI reference model and partial Bluetooth protocol stack

The following six sections describe the functions of each layer of the Bluetooth protocols used by the BlueMeter Hub.

3.1 RF Layer

The Bluetooth radio performs spectrum spreading by frequency hopping in 79 frequencies that are displaced by 1 MHz, between 2.402GHz and 2.480GHz. The maximum transmit power is <12 dBm, but the actual transmitted power on a link may be lower for optimization.

The modulation uses GFSK (Gaussian Frequency Shift Keying) where a binary one is represented by a positive frequency deviation and a binary zero by a negative frequency deviation. The BT (Time-Bandwidth product) is 0.5 and the modulation index is between 0.28 and 0.35.

The frequency synthesizer changes frequency between receive slot and transmit slot, but always returns to the same transmit frequency. The transmitted initial center frequency accuracy must be ±75 kHz from center.

The receiver has a sensitivity level of -70dBm or better to achieve a bit error rate (BER) of 0.1% or better. The interference performance on co-channel and adjacent 1 MHz and 2 MHz are measured with the wanted signal 10 dB over the reference sensitivity level. On all other frequencies, the wanted signal is 3 dB over the reference sensitivity level. The out of band blocking is measured with the wanted signal 3 dB over the reference sensitivity level. The interfering signal is a continuous wave signal. The BER is less than or equal to 0.1%.

The reference sensitivity performance, BER = 0.1%, is met under the following conditions:

- The wanted signal at frequency f_0 with a power level 6 dB over the reference sensitivity level;

- A static sine wave signal at f_1 with a power level of -39 dBm;
- A Bluetooth modulated signal at f_2 with a power level of -39 dBm; and
- $f_0 = 2f_1 - f_2$ and $|f_2 - f_1| = n$ MHz, where n can be 3, 4, or 5.

3.2 Baseband Layer

The baseband is responsible for hop selection, low-level timing control, channel coding and decoding, and management of the link within the domain of a single data packet transfer.

The frequency channel on which to transmit is chosen among the 79 frequencies according to a pseudo-random hopping sequence. Two or more devices using the hopping sequence form a piconet. There is one master and one or more slaves in each piconet. The sequence is unique to the piconet and is determined by the Bluetooth device address of the master; the phase in the hopping sequence is determined by the Bluetooth clock of the master.

The channel is divided into time slots. Each slot is 625 µs in length and corresponds to a hop frequency. Specifically, the time slots are numbered according to the Bluetooth clock of the piconet master. A time division duplexing (TDD) scheme is used where master and slave alternatively transmit. The master starts its transmission in even-numbered time slots only, and the slave starts its transmission in odd-numbered time slots only. The packet start is aligned with the slot start.

The baseband handles two types of links: Synchronous Connection-Oriented (SCO) and Asynchronous Connection-Less (ACL) link. The SCO link is a symmetric point-to-point link between a master and a single slave in the piconet. The master maintains the SCO link by using reserved slots at regular intervals. The ACL link is a point-to-multipoint link between the master and all the slaves participating on the piconet. In the slots not reserved for the SCO links, the master can establish an ACL link on a per-slot basis to any slave, including the slave already engaged in an SCO.

All data on the piconet channel is conveyed in packets. There are 13 different packet types defined for the baseband layer. All higher layers use these packets to compose higher-level PDU's (protocol data units). Each packet consists of three entities, the access code (68 or 72 bits), the header (54 bits), and the payload (0-2745 bits). The access code is used for timing synchronization, offset compensation, paging and inquiry. The header contains information for packet acknowledgement, packet numbering for out-of-order packet reordering, flow control, slave address and error check for header. The packet payload can contain either voice field, data field or both. If it has a data field, the payload will also contain a payload header. The coding applied to the data bit-stream consists of

- Cyclic Redundancy Check (CRC) and Header Error Check (HEC);
- Encryption;
- Whitening or bit randomization; and
- Forward Error Correction (FEC).

Whitening and the HEC are applied to all packets with at least a header. FEC and CRC are only applied to certain packets.

3.3 Link Controller Layer

The link controller operates in two major states, Standby and Connection. There are also seven substates to add slaves or make connections; these are page, page scan, inquiry, inquiry scan, master response, slave response and inquiry response.

The Standby state is the default low power state. Only the native clock is running and there is no interaction with any device. In the Connection state, the master and slave can exchange packet, using the channel access code and the master Bluetooth clock. The hopping scheme used is the channel hopping scheme.

Normally, a connection between two devices occurs in the following fashion: If nothing is known about a remote device, both the inquiry and page procedure have to be followed. If some details are known about a remote device, only the paging procedure is needed

The inquiry procedure enables a device to discover which devices are in range, and determine the addresses and clocks for the devices. The inquiry procedure involves a unit (the source) sending out inquiry packets (inquiry state) and then receiving the inquiry reply. The unit that receives the inquiry packets (the destination), will hopefully be in the inquiry scan state to receive the inquiry packets. The destination will then enter the inquiry response state and send an inquiry reply to the source.

With the paging procedure, an actual connection can be established. The paging procedure typically follows the inquiry procedure. Only the Bluetooth device address is required to set up a connection. Knowledge about the clock (clock estimate) will accelerate the setup procedure. A unit that establishes a connection will carry out a page procedure and will automatically be the master of the connection.

There are four modes in the Connection state: active, hold, sniff and park. In the active mode, the unit actively participates on the channel. In the hold, sniff, and park mode, a slave device listens to the piconet at reduced rate, thus reducing its duty cycle.

Each piconet has a different master, and its channel hopping sequence and phase is determined by the master. In addition, the packets carried on the channels are preceded by different channel access codes as determined by the master device addresses.

If multiple piconets cover the same area, a unit can participate in two or more overlaying piconets by applying time multiplexing. To participate on the proper channel, it should use the associated master device address and proper clock offset to obtain the correct phase. A unit can act as a slave in several piconets, but only as a master in a single piconet. A group of piconets in which connections consists between different piconets is called a scatternet

An existing master and slave may swap roles in two steps: first a TDD slot switch, followed by a hop sequence switch. Then, if so desired, other slaves of the old piconet can be transferred to the new piconet.

3.4 Link Manager

The Link Manager (LM) carries out link setup, authentication, link configuration and other protocols. It discovers other remote LM's and communicates with them via the Link Manager Protocol (LMP). To perform its service provider role, the LM uses the services of the underlying Link Controller (LC). In particular, the LM carries out the following operations

- Attaching slaves to a piconet, and allocating their active member addresses;
- Breaking connections to detach slaves from a piconet;
- Configuraing the link including controlling master/slave switches;
- Establishing ACL and SCO links;
- Putting connections into low-power modes (sniff, hold & park); and
- Controlling test modes.

The Link Manager Protocol essentially consists of a number of PDUs, which are sent from one device to another, determined by the packet header. LM PDUs are always sent as single-slot packets and the payload header is therefore one byte.

3.5 Host Controller Interface Layer

The host controller interface (HCI) provides a command interface to the link manager, and provides access to hardware status and control registers. Essentially, this interface provides a uniform method of accessing the baseband capabilities.

The HCI firmware is located on the Host Controller, the actual Bluetooth hardware device. The HCI firmware implements the HCI Commands for the Bluetooth hardware by accessing baseband commands, link manager commands, hardware status registers, control registers, and event registers.

HCI Driver is located on the Host, an HCI-enabled software entity. The Host will receive asynchronous notifications of HCI event; HCI events are used for notifying the Host when something occurs. When the Host discovers that an event has occurred, it will then parse the received event packet to determine which event occurred.

The HCI Driver and Firmware communicate via the Host Controller Transport Layer, i.e. a definition of the several layers that may exist between the HCI driver on the host system and the HCI firmware in the Bluetooth hardware. Several different Host Controller Layers can be used, of which 3 have been defined initially for Bluetooth: USB, UART and RS232. The Host receives asynchronous notifications of HCI events independent of which Host Controller Transport Layer is used.

3.6 Logical Link Control and Adaptation Protocol Layer

The Logical Link Control and Adaptation Layer Protocol (L2CAP) provides connection-oriented and connectionless data services to upper layer protocols with protocol multiplexing capability, segmentation and reassembly operation, and group abstractions. L2CAP permits higher level protocols and applications to transmit and receive L2CAP data packets up to 64 kilobytes in length. L2CAP is only defined for ACLs.

L2CAP supports protocol multiplexing because the baseband protocol does not support any *type* field identifying the higher layer protocol being multiplexed above it. L2CAP is able to distinguish between upper layer protocols.

Due to the baseband payload limits, large L2CAP packets must be segmented into multiple smaller baseband packets prior to their transmission over the air. Similarly, multiple received baseband packets may be reassembled into a single larger L2CAP packet following a simple integrity check.

The L2CAP connection establishment process allows the exchange of information regarding the quality of service (QoS) expected between two Bluetooth units. Each L2CAP implementation must monitor the resources used by the protocol and ensure that QoS contracts are honored.

The baseband protocol supports the concept of a piconet, a group of devices synchronously hopping together using the same clock. The L2CAP group abstraction permits implementations to efficiently map protocol groups on to piconets.