BLUETOOTH® DOC	Date / Year-Month-Day 2011-05-24	Approved Adopted	Revision V10r00	Document No HTS_SPEC
Prepared By MED WG	E-mail Address med-feedback@bluetooth.org			N.B.

HEALTH THERMOMETER SERVICE

Abstract:

This service exposes temperature and other data from a Thermometer intended for healthcare and fitness applications.

Revision History

Revision	Date (yyyy-mm-dd)	Comments
D09r00	2010-11-22	Initial Draft from Health Thermometer UCRDD.
D09r04	2010-12-08	Version of spec used for IOP.
D09r05	2010-12-16	Approved by BARB.
V10r00	2011-05-24	Adopted by the Bluetooth SIG Board of Directors

Contributors

Name	Company
Sen-Der Huang	Broadcom
Victor Zhodzishsky	Broadcom
Joe Decuir	CSR
Robin Heydon	CSR
Nick Hunn	CSR
Robert Hughes	Intel Corporation
Anindya Bakshi	MindTree Limited
Krishna Shingala	MindTree Limited
Dennis McCain	Murata
Tetsu Nishimura	Murata
Torstein Heggebø	Nordic Semiconductor
Brian Redding	Qualcomm
Jason Hillyard	Wicentric

Disclaimer and Copyright Notice

The copyright in this specification is owned by the Promoter Members of *Bluetooth*® Special Interest Group (SIG), Inc. ("*Bluetooth* SIG"). Use of these specifications and any related intellectual property (collectively, the "Specification"), is governed by the Promoters Membership Agreement among the Promoter Members and *Bluetooth* SIG (the "Promoters Agreement"), certain membership agreements between *Bluetooth* SIG and its Adopter and Associate Members (the "Membership Agreements") and the *Bluetooth* Specification Early Adopters Agreements (1.2 Early Adopters Agreements) among Early Adopter members of the unincorporated *Bluetooth* SIG and the Promoter Members (the "Early Adopters Agreement"). Certain rights and obligations of the Promoter Members under the Early Adopters Agreements have been assigned to *Bluetooth* SIG by the Promoter Members.

Use of the Specification by anyone who is not a member of *Bluetooth* SIG or a party to an Early Adopters Agreement (each such person or party, a "Member"), is prohibited. The legal rights and obligations of each Member are governed by their applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement. No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Any use of the Specification not in compliance with the terms of the applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement is prohibited and any such prohibited use may result in termination of the applicable Membership Agreement or Early Adopters Agreement and other liability permitted by the applicable agreement or by applicable law to *Bluetooth* SIG or any of its members for patent, copyright and/or trademark infringement.

THE SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, SATISFACTORY QUALITY, OR REASONABLE SKILL OR CARE, OR ANY WARRANTY ARISING OUT OF ANY COURSE OF DEALING, USAGE, TRADE PRACTICE, PROPOSAL, SPECIFICATION OR SAMPLE.

Each Member hereby acknowledges that products equipped with the *Bluetooth* technology ("*Bluetooth* products") may be subject to various regulatory controls under the laws and regulations of various governments worldwide. Such laws and regulatory controls may govern, among other things, the combination, operation, use, implementation and distribution of *Bluetooth* products. Examples of such laws and regulatory controls include, but are not limited to, airline regulatory controls, telecommunications regulations, technology transfer controls and health and safety regulations. Each Member is solely responsible for the compliance by their *Bluetooth* Products with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their *Bluetooth* products related to such regulations within the applicable jurisdictions. Each Member acknowledges that nothing in the Specification provides any information or assistance in connection with securing such compliance, authorizations or licenses**. NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS.**

ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE SPECIFICATION IS EXPRESSLY DISCLAIMED. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST *BLUETOOTH* SIG AND ITS PROMOTER MEMBERS RELATED TO USE OF THE SPECIFICATION.

Bluetooth SIG reserve the right to adopt any changes or alterations to the Specification as it deems necessary or appropriate.

Document Terminology

The Bluetooth SIG has adopted Section 13.1 of the IEEE Standards Style Manual, which dictates use of the words ``shall'', ``should'', ``may'', and ``can'' in the development of documentation, as follows:

The word *shall* is used to indicate mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).

The use of the word *must* is deprecated and shall not be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

The use of the word *will* is deprecated and shall not be used when stating mandatory requirements; *will* is only used in statements of fact.

The word *should* is used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain course of action is deprecated but not prohibited (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

Table of Contents

1 Introduction

The Health Thermometer Service exposes temperature and other data related to a thermometer used for healthcare applications.

1.1 Conformance

If a device claims conformance to this service, all capabilities indicated as mandatory for this service shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional capabilities for which support is indicated, are subject to verification as part of the Bluetooth qualification program.

1.2 Service Dependency

This service is not dependent upon any other services.

1.3 Bluetooth Specification Release Compatibility

This specification is compatible with any Bluetooth core specification [1] that includes the Generic Attribute Profile (GATT) specification and the Bluetooth Low Energy Controller specification.

1.4 GATT Sub-Procedure Requirements

Requirements in this section represent a minimum set of requirements for a Thermometer (Server). Other GATT sub-procedures may be used if supported by both Client and Server.

The table below summarizes additional GATT sub-procedure requirements beyond those required by all GATT Servers.

GATT Sub-Procedure	Server Requirements
Write Characteristic Value	C1
Notifications	C2
Indications	M
Read Characteristic Descriptors	M
Write Characteristic Descriptors	M

Table 1.1: GATT Sub-Procedure Requirements

C1: Mandatory if the Write property for the Measurement Interval characteristic is supported, otherwise excluded for this service.

C2: Mandatory if the Intermediate Temperature characteristic is supported, otherwise excluded for this service.

1.5 Transport Dependencies

This service shall operate over LE transport only. For BR/EDR (and HS) the Health Device Profile [2] is to be used.

1.6 Error Codes

This service defines the following Attribute Protocol Application Error codes:

Name	Error Code	Description
Out of Range	0x80	The value is considered invalid and outside of the range allowed by the characteristic.

1.7 Byte Transmission Order

All characteristics used with this service shall be transmitted with the least significant octet first (i.e., little endian). The least significant octet is identified in the characteristic definitions in [3].

2 Service Declaration

The Health Thermometer Service shall be instantiated as a «Primary Service».

The service UUID shall be set to «Health Thermometer Service». The UUID value assigned to «Health Thermometer Service» is defined in [3].

3 Service Characteristics

The following characteristics are exposed in the Health Thermometer Service. Unless otherwise specified, only one instance of each characteristic is permitted within this service.

Characteristic Name	Requirement	Mandatory Properties	Optional Properties	Security Permissions
Temperature Measurement	M	Indicate		None.
Client Characteristic Configuration descriptor	M	Read, Write		None.
Temperature Type	O	Read		None.
Intermediate Temperature	O	Notify		None.
Client Characteristic Configuration descriptor	C.1	Read, Write		None.
Measurement Interval	O	Read	Indicate, Write	Read: None. Writable with authentication.
Client Characteristic Configuration descriptor	C.2	Read, Write		None.
Valid Range descriptor	C.3	Read		None.

Table 3.1: Health Thermometer Service characteristics

C.1: Mandatory if Intermediate Temperature characteristic is supported, otherwise excluded.

C.2: Mandatory if Measurement Interval supports indications, otherwise excluded.

C.3: Mandatory if Measurement Interval is supported and Writable, otherwise excluded.

Notes:

- Security Permissions of “None” means that this service does not impose any requirements.
- Properties not listed as Mandatory or Optional are Excluded.

3.1 Temperature Measurement

The Temperature Measurement characteristic is used to send a temperature measurement. Included in the characteristic are a Flags field (for showing the units of temperature and presence of optional fields), the temperature measurement value and, depending upon the contents of the Flags field, the time of that measurement and the temperature type.

The Temperature Type field in the Temperature Measurement characteristic is intended to be used when the value of temperature type is non-static (e.g., configured using a switch or a simple user interface on a thermometer). For temperature type values that are static, the Temperature Type characteristic (see Section 3.2) should be used.

3.1.1 Characteristic Behavior

When the *Client Characteristic Configuration* descriptor is configured for indication and a temperature measurement is available, this characteristic shall be indicated while in a connection.

If a temperature measurement is available and a connection is not currently established, the Server shall become connectable to allow the Collector to create a link.

The Temperature Measurement characteristic contains time-sensitive data, thus the requirements for time-sensitive data and data storage defined in Section 3.5 apply.

3.1.1.1 Flags Field

The Flags field shall be included in the Temperature Measurement characteristic.

Reserved for Future Use (RFU) bits in the Flags field shall be set to 0.

3.1.1.2 Temperature Measurement Value Field

This Temperature Measurement Value field shall be included in the Temperature Measurement characteristic.

If the unit of the Temperature Measurement is in Celsius, bit 0 of the Flags field shall be set to 0. Otherwise, the unit shall be Fahrenheit and bit 0 of the Flags field shall be set to 1.

The Temperature Measurement Value field may contain special float value NaN (0x007FFFFF) defined in IEEE 11073-20601 [4] to report an invalid result from a computation step or missing data due to the hardware's inability to provide a valid measurement.

3.1.1.3 Time Stamp Field

The Time Stamp field shall be included in the Temperature Measurement characteristic if the device supports storing of data. Otherwise it is optional.

If a time stamp is supported, the Server shall set bit 1 of the Flags field to 1 and include the Time Stamp field in the Temperature Measurement characteristic. Otherwise bit 1 of the Flags field shall be set to 0 and the Time Stamp field shall not be included.

The value of the Time Stamp field is derived from a source of date and time at the time of measurement. If the Time Stamp feature is supported, a source of date and time is mandatory.

The date and time of the device may be updated by various means such as via a simple user interface on the device, via an external time service, etc.

The time stamp field is defined to use the same format as the Date Time characteristic [3]; however, a value of 0 for the month or day fields shall not be used for this service.

3.1.1.4 Temperature Type Field

If the Temperature Type value is supported and is non-static, the Server shall set bit 2 of the Flags field to 1 and include the Temperature Type field in the Temperature

Measurement characteristic. Otherwise bit 2 of the Flags field shall be set to 0 and the Temperature Type field shall not be included.

3.1.2 Characteristic Descriptors

3.1.2.1 Client Characteristic Configuration Descriptor

The *Client Characteristic Configuration* descriptor shall be included in the Temperature Measurement characteristic.

3.2 Temperature Type

The Temperature Type characteristic is one of two methods that are used to describe the type of temperature measurement in relation to the location on the human body at which the temperature was measured.

There are two exclusive methods to enable a Thermometer to provide temperature type information to a Collector. Either one method or the other is used, but not both. The Temperature Type characteristic is intended to be supported when the value is static while in a connection. For temperature type values that are non-static while in a connection (e.g. configured using a switch or a simple user interface on a thermometer), the Temperature Type characteristics shall not be supported, rather the Temperature Type field in the Temperature Measurement characteristic (refer to Section 3.1) shall be supported.

If the thermometer is for general use, a value of 0x02 Body (general) may be used.

3.2.1 Characteristic Behavior

The Temperature Type characteristic returns the current temperature type value when read.

3.3 Intermediate Temperature

The Intermediate Temperature characteristic is used to send intermediate temperature values to a device for display purposes while the measurement is in progress.

3.3.1 Characteristic Behavior

The Intermediate Temperature characteristic is notified frequently during the course of a measurement so that a receiving device can effectively update the display on its user interface during the measurement process. The update rate is defined by the Server and should not be greater than a rate necessary to provide an acceptable user display on the Client. Typical update intervals range from 0.25 seconds to 2 seconds.

When the *Client Characteristic Configuration* descriptor is configured for notification and an intermediate temperature value is available, this characteristic shall be notified while in a connection.

If an intermediate temperature value is available and a connection is not currently established, the Server shall become connectable to allow the Collector to create a link.

To avoid transmitting unnecessary data, the Time Stamp and Temperature Type fields should not be used in the Intermediate Temperature characteristic.

Once the measurement process is complete and a stable temperature measurement is available, the Server shall stop notifications of the Intermediate Temperature characteristic and, if indications are enabled, shall indicate the Temperature Measurement characteristic.

3.3.1.1 Flags Field

The Flags field shall be included in the Intermediate Temperature characteristic.

RFU bits in the Flags field shall be set to 0.

3.3.1.2 Intermediate Temperature Value Field

This Intermediate Temperature Value field shall be included in the Intermediate Temperature characteristic.

If the unit of the Intermediate Temperature is in Celsius, bit 0 of the Flags field shall be set to 0. Otherwise, the unit shall be Fahrenheit and bit 0 of the Flags field shall be set to 1.

The Intermediate Temperature Value field may contain special float value NaN (0x007FFFFF) defined in IEEE 11073-20601 [4] to report an invalid result from a computation step or missing data due to the hardware’s inability to provide a valid measurement.

3.3.1.3 Time Stamp Field

The Time Stamp field may be included in the Intermediate Temperature characteristic.

If a time stamp is supported, the Server shall set bit 1 of the Flags field to 1 and include the Time Stamp field in the Intermediate Temperature characteristic. Otherwise bit 1 of the Flags field shall be set to 0 and the Time Stamp field shall not be included.

The value of the Time Stamp field shall be derived from the date and time of the device at the time of measurement.

The date and time of the device may be updated by various means such as via a simple user interface on the device, via an external time service, etc.

The time stamp field is defined to use the same format as the Date Time characteristic [3], however, a value of 0 for the month or day fields is not permitted for this service.

3.3.1.4 Temperature Type Field

If the Temperature Type value is supported, the Server shall set bit 2 of the Flags field to 1 and include the Temperature Type field in the Intermediate Temperature characteristic. Otherwise bit 2 of the Flags field shall be set to 0 and the Temperature Type field shall not be included.

3.3.2 Characteristic Descriptors

3.3.2.1 Client Characteristic Configuration Descriptor

The *Client Characteristic Configuration* descriptor shall be included in the Intermediate Temperature characteristic.

3.4 Measurement Interval

The Measurement Interval characteristic is used to enable and control the interval between consecutive temperature measurements. Note: This interval is not related to the intermediate temperature feature described in Section 3.3 and is only applicable to the Temperature Measurement characteristic.

The Valid Range descriptor defined in Section 3.4.2.2 enables a Client to determine the supported range of the Measurement Interval characteristic value.

3.4.1 Characteristic Behavior

The Measurement Interval characteristic returns the measurement interval value when read.

If writable, the Measurement Interval characteristic sets the measurement interval value when written.

If the *Client Characteristic Configuration* descriptor is configured for indication and a new measurement interval is available, this characteristic can be indicated while in a connection.

If the value of the Measurement Interval characteristic is greater than zero and the *Client Characteristic Configuration* descriptor of the Temperature Measurement characteristic is configured for indications, the Server shall make temperature measurements at intervals according to the value of the Measurement Interval characteristic and indicate the Temperature Measurement characteristic as described in Section 3.1.

If the Measurement Interval characteristic is set to 0, no periodic measurement shall be made.

If the Client attempts to write a value to the Measurement Interval characteristic that cannot be supported by the Server, the Server shall send an error response with the error code set to *Out of Range*.

When a new non-zero value is written, the Server shall reset any existing measurement interval and start the measurement process. If a measurement is already in progress during the reset, the results of the measurement underway may be used in the Temperature Measurement characteristic.

If the value of Measurement Interval is changed locally on the Server and the *Client Characteristic Configuration* descriptor of this characteristic is configured for indication, the Server shall not become connectable to allow the Collector to create a link. Rather, the Server shall indicate the current value of the characteristic the next time there is a connection.

3.4.2 Characteristic Descriptors

3.4.2.1 Client Characteristic Configuration Descriptor

If the properties of this characteristic allow indications, then the *Client Characteristic Configuration* descriptor shall be included in the Measurement Interval characteristic, otherwise this descriptor shall not be included.

3.4.2.2 Valid Range Descriptor

This descriptor is Mandatory if the Measurement Interval characteristic is Writable.

The Valid Range descriptor enables a Client to determine the supported range of the Measurement Interval characteristic value.

The Valid Range descriptor returns the range of supported measurement interval values (i.e., the lower inclusive and upper inclusive values) when read.

A value of 0 is not valid for the lower inclusive value of the descriptor for this profile.

When used with the Measurement Interval characteristic, the Valid Range descriptor contains two unsigned 16-bit integers representing the valid range of values that the Measurement Interval characteristic can support in units of seconds.

See Section 3.6 for further requirements related to this descriptor.

3.5 Requirements for Time-Sensitive Data

The Temperature Measurement characteristic contains time sensitive data and is considered a time-sensitive characteristic, thus the following requirements apply:

If the time stamp feature is not supported:

> The value of the time-sensitive characteristic shall be discarded if either the connection does not get established or if the indication is not successfully acknowledged by the Client during the connection.

If the time stamp feature is supported:

> The value of the time-sensitive characteristic is strongly recommended to be stored if either the connection does not get established or if the indication is not successfully acknowledged by the Client during the connection.
>
> The Server shall attempt to send stored data to the Client during the next connection.
>
> If data has been successfully transferred by a Server and acknowledged by the Client, the data shall not be retransmitted.
>
> For Servers supporting the sending of the Temperature Measurement characteristic at intervals using the Measurement Interval characteristic, it is recommended that at least one day of time-stamped data can be stored. If the device is configured to send data at intervals of 30 minutes over a 24-hour period, this equates to a recommendation of at least 48 stored data measurements. For more basic scenarios, the storing of at least 30 data measurements is recommended.

If the maximum storage capacity in the Server is reached, the Server should purge the oldest data to make room for the most recent data.

When transmitting stored data, the oldest data shall be sent first followed by the next oldest data (in FIFO order) until all stored data has been transferred.

The Server may attempt to send stored measurements any time there is a connection including when measurements are taken at periodic intervals (see Section 3.4).

3.6 Valid Range

The following section contains the generic description of the Valid Range characteristic descriptor. This section may be removed from future versions of this document and moved into another document as the descriptor may be used in other profiles and services. The specific use and requirements for this descriptor are contained in earlier sections of this specification.

The *Valid Range* declaration is an optional characteristic descriptor that defines the valid range of values for the *Characteristic Value*. The descriptor is used to communicate the valid range of values that can be written to a *Characteristic Value* by a client. The characteristic descriptor may occur in any position within the characteristic definition after the *Characteristic Value*. Only one *Valid Range* declaration shall exist in a characteristic definition.

The characteristic descriptor is contained in an Attribute and the *Attribute Type* shall be set to the UUID for ‹‹Valid Range›› and the *Attribute Value* shall be set to the characteristic descriptor value. The *Attribute Permissions* are specified by the profile or may be implementation specific if not specified otherwise.

Attribute Handle	**Attribute Type**	**Attribute Value**		**Attribute Permissions**
0xNNNN	0xuuuu – UUID for ‹‹Valid Range››	Minimum Value	Maximum Value	Higher layer profile or implementation specific

Table 3.2: Valid Range Declaration

The definition of the *Valid Range* descriptor *Attribute Value* is the following:

Field Name	**Value Size**	**Description**
Minimum Value	Size of the Characteristic Value	Minimum valid Characteristic Value
Maximum Value	Size of the Characteristic Value	Maximum valid Characteristic Value.

Table 3.3: Valid Range Definition

The minimum and maximum *Characteristic Values* are inclusive to the values specified in the Minimum Value and Maximum Value fields of the descriptor. *Characteristic Values* outside the valid range defined by this descriptor are considered invalid values and an error condition. The size of the *Minimum Value* and *Maximum Value* defined in the descriptor are based on the size of the *Characteristic Value* described by the descriptor.

The value of the *Maximum Value* shall always be greater than or equal to the value of the *Minimum Value*.

The units and data type of the Minimum Value and Maximum Value fields of the descriptor shall be the same as the characteristic.

3.6.1 Bit Ordering

The bit ordering used for the *Valid Range* descriptor shall be little-endian.

4 Acronyms and Abbreviations

Acronyms and Abbreviations	Meaning
BR/EDR	Basic Rate / Enhanced Data Rate
GAP	Generic Access Profile
GATT	Generic Attribute Profile
HS	High Speed
LE	Low Energy
NaN	Not a Number
RFU	Reserved for Future Use
UUID	Universally Unique Identifier

Table 4.1: Acronyms and Abbreviations

5 References

[1] Bluetooth Core Specification v4.0
[2] Health Device Profile v1.0
[3] Characteristic and Descriptor descriptions are accessible via the Bluetooth SIG Assigned Numbers.
[4] IEEE Std 11073-20601™- 2008 Health Informatics - Personal Health Device Communication - Application Profile - Optimized Exchange Protocol - version 1.0 or later