10 May 2011

Status
Approved

Revision
V10r00

Document File Name
Personal Health Devices WP_V10.doc

Document Owner
Medical Devices WG

E-mail Address
med-feedback@bluetooth.org

Bluetooth SIG Confidential

Bluetooth®
SPECIAL INTEREST GROUP

PERSONAL HEALTH DEVICES TRANSCODING WHITE PAPER

ABSTRACT: This document is informative. It aims to facilitate the task of implementing a transcoder from GATT based specifications designed for *Bluetooth* LE devices to a format compatible with IEEE 11073-20601. It provides recommendations and examples describing how a transcoding process can be done.

Revision History		
Revision	**Date**	**Description**
D00r01	21 December 2010	Initial draft.
D09r01-05	26 January 2011	Incorporated feedback from MED WG reviews.
D09r07	30 March 2011	Prepared for review by all SIG members.
D10r00-02	26 April 2011	Rolled revision number to 1.0. Prepared for BARB review.
D10r03	05 May 2011	Editorial review.
V10r00	10 May 2011	Approved by the Bluetooth Board of Directors.

Contributors	
Name	**Company**
Raúl González Gómez	AT4 wireless
Robert D. Hughes	Intel Corporation
Doug Bogia	Intel Corporation
Krishna Shingala	MindTree
Kurt Kermes	Nonin Medical
Mateus Lima	Signove
Raul Herbster	Signove
Elvis Pfutzenreuter	Signove
Len Ott	Socket Mobile
John R. Barr	Motorola Mobility, Inc.

Contents

1. Overview

It is the design intent of this document to describe how data sent by a *Bluetooth*® low energy (LE) Sensor and received by a Collector can be transcoded at the Collector into IEEE Std. 11073-20601a [1] compatible nomenclature and model, thus enabling compatibility with the ISO/IEEE Std. 11073-104xx family of standards (both hereafter known as IEEE). *Bluetooth* Sensor profiles for *Bluetooth* LE implementations have been developed using the GATT based profile architecture designed to support low power and low cost device implementation. Data values used in these profiles are defined as characteristics associated with a UUID defined on the Bluetooth SIG assigned numbers website [4].

At the simplest level this document covers how characteristic values can be mapped or transcoded in a consistent way to IEEE nomenclature/object/attribute equivalents. This data compatibility will enable data from *Bluetooth* LE devices to be used in the broader health ecosystem such that the transcoded measurement data will look the same as data from a Health Device Profile or a USB Personal Healthcare Device Class for a given device.

This data compatibility will enable data to be useable and consumable by a variety of healthcare-related organizations including the Continua Health Alliance and standards organizations related to health records such as HL7. All mandatory (as well as some optional) attributes defined for each specialization in IEEE are supported by *Bluetooth* profiles defined within this document, but support for optional attributes is not specifically required for compatibility.

For the profiles encompassed in this document, all characteristics and fields that are relevant to IEEE have been defined with the intent that they can be transcoded at the Collector without any loss of precision. Should a discrepancy be discovered, the Bluetooth SIG plans to correct any incompatibilities. In order to enable such a process for a particular device, the Collector device implementing transcoding software is required to follow the general requirements in Section 3 in addition to the device specific requirements in Section 3. Section 3 will be expanded as new *Bluetooth* profiles become available. Section 4 provides an end-to-end example describing how data can be mapped from a LE Health Thermometer to a Collector implementing a Transcoder.

While it is beyond the scope of this document to mandate or specify a specific *method* for transcoding *Bluetooth* characteristics into IEEE, this white paper provides requirements and guidelines to enable implementations to do so. This document does not discuss IEEE concepts and details, rather focuses on how data from *Bluetooth* sensors can be transcoded for use in the IEEE domain.

Although some areas of this document summarize requirements of IEEE documents as a useful reference, refer to the IEEE standards to ensure the most accurate information regarding requirements associated with IEEE specifications. Similarly, refer to the relevant *Bluetooth* profile specifications with their associated service specifications and characteristic(s) as the official sources for *Bluetooth* related requirements.

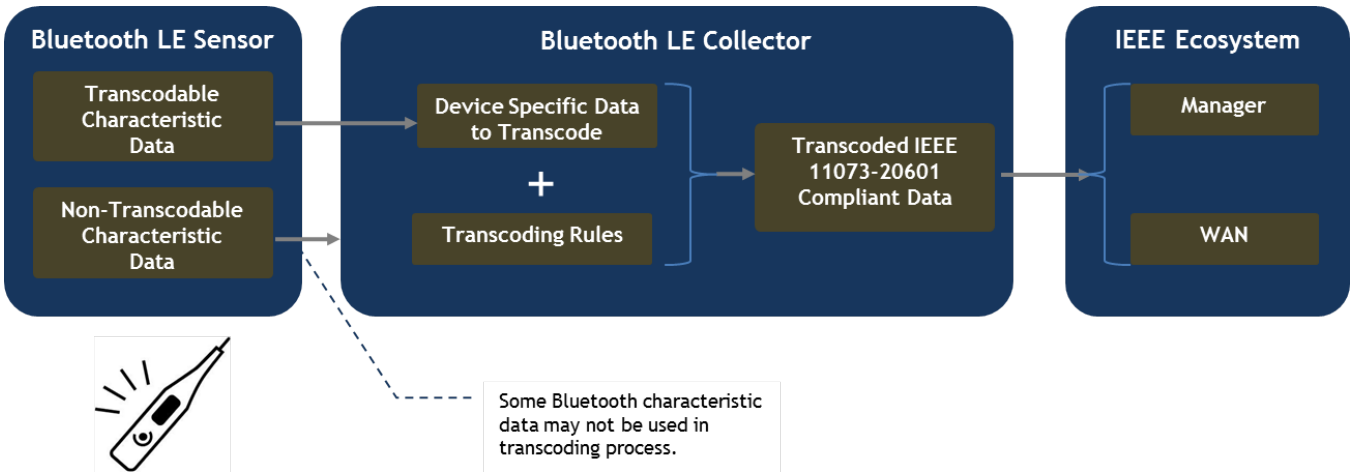

Figure 1: Bluetooth LE Sensor to IEEE Data Flow

2. General Data Requirements

This section describes general data mapping requirements that are common to all devices encompassed by this white paper. This section describes the transcoding of general data from a *Bluetooth* sensor device for compatibility with the IEEE ecosystem. Device-specific data requirements are described in Section 3.

2.1 COMMON MDS CLASS REQUIREMENTS

This section describes general Medical Device System (MDS) class requirements. IEEE attributes not mentioned in this section likely refer to *Bluetooth* device specific data requirements and are shown in Section 3.

The Device Information Service (DIS) [2] is a general *Bluetooth* service designed to describe characteristics that are often common between different Sensors. The DIS includes characteristics that contain information such as the manufacturer name, model number, hardware revision, firmware revision and software revision among others. Many characteristics in the DIS are used within the IEEE and Continua Health Alliance infrastructure.

IEEE attribute	*Bluetooth* equivalent characteristic	*Bluetooth* Service	*Bluetooth* data type	IEEE attribute type (ASN.1)	IEEE data type (informative)
Handle	None	N/A	N/A	HANDLE[1]	INT-U16
System-Model	Model Number String, Manufacturer Name String	DIS	UTF-8 String, UTF-8 String	SystemModel[2]	(OCTET STRING, OCTET STRING)
System-Id	System ID	DIS	EUI-64	OCTET STRING	OCTET STRING
Attribute-Value-Map	None	N/A	N/A	AttrValMap[3]	List of (INT-U16, INT-U16)
Production-Specification	Serial Number String, Hardware Revision String, Software Revision String, Firmware Revision String[4]	DIS	UTF-8 String	ProductionSpec	List of (INT-U16, INT-U16, OCTET STRING)
Date-and-Time	Date Time	Various[5]	Opaque Structure	AbsoluteTime	(INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8)
Mds-Time-Info	None	N/A	N/A	MdsTimeInfo	(BITS-16, INT-U16, INT-U32, INT-U16, INT-U16, INT-U32)
Relative-Time	None	N/A	N/A	RelativeTime	INT-U32
HiRes-Relative-Time	None	N/A	N/A	HighResRelativeTime	OCTET STRING (SIZE(8))
Date-and-Time-Adjustment	None	N/A	N/A	AbsoluteTimeAdjust	OCTET STRING (SIZE(6))
Power-Status	Drafting	Drafting	N/A	PowerStatus	BITS-16
Battery-Level	Drafting	Drafting	N/A	INT-U16	INT-U16
Remaining-Battery-Time	None	N/A	N/A	BatMeasure	(FLOAT-Type, INT-U16)
Reg-Cert-Data-List	IEEE 11073-20601 Regulatory Certification Data List	DIS	Opaque Structure	RegCertDataList	List of ((INT-U8, INT-U8), ANY)
System-Type-Spec-List	None	N/A	N/A	TypeVerList[6]	List of (INT-U16, INT-U16)
Confirm-Timeout	None	N/A	N/A	RelativeTime	OCTET STRING (SIZE(6))

Table 1: Common MDS Class Requirements

Notes:

1. Always set to 0.
2. SystemModel is described as SEQUENCE(Manufacturer Name, Model Number).
3. This IEEE attribute is transmission-related only. As this document is only relevant for nomenclature and model compatibility and not mandating any way to reach the IEEE domain, this attribute is implementation specific.
4. Each field is mapped into a unique entry of ProductionSpec list, such as Serial Number String is mapped as (0x0001, INT-U16, serial_number_value). See Section 2.2.4.
5. Provided by any service that includes a Time Stamp.
6. For each profile encompassed in this white paper that is implemented by the device, an entry has to be added to the TypeVerList. For example, if a device implements the Health Thermometer Profile [3], the following entry is required to be added to the TypeVerList:

   0x10 0x08 type = MDC_DEV_SPEC_PROFILE_TEMP
   0x00 0x01 version = version 1 of the specialization

2.2 TRANSCODING *BLUETOOTH* CHARACTERISTICS TO IEEE ATTRIBUTES

2.2.1 FLOATING POINT TYPE (FLOAT-TYPE) DATA TYPE

The following information is defined in ISO/IEEE Std. 11073-2060™1-2008 [1].

The FLOAT-Type data type is defined to represent numeric values that are not integer in type. The FLOAT-Type is defined as a 32-bit value with a 24-bit mantissa and an 8-bit exponent. See Annex F.6 of [1] for a thorough definition of the FLOAT-Type. This data type is defined as follows:

	Exponent	**Mantissa**
Size	1 octet	3 octets

--
-- 32-bit float type; the integer type is a placeholder only
--
FLOAT-Type ::= INT-U32

The 32 bits contain an 8-bit signed exponent to base 10, followed by a 24-bit signed integer (mantissa).

Special values are assigned to express the following:

- + INFINITY [exponent 0, mantissa +(2^23 –2) → 0x007FFFFE]
- NaN (not a number) [exponent 0, mantissa +(2^23 –1) → 0x007FFFFF]
- NRes (not at this resolution) [exponent 0, mantissa –(2^23) → 0x00800000]
- Reserved for future use [exponent 0, mantissa –(2^23–1) → 0x00800001]
- – INFINITY [exponent 0, mantissa –(2^23 –2) → 0x00800002]

NaN is used to report an invalid result from a computation step or to indicate missing data due to the hardware's inability to provide a valid measurement, perhaps from sensor disturbances.

NRes is used to report that the value cannot be represented with the available range and resolution, possibly resulting from an overflow or underflow situation.

Example

Example for temperature measurement in IEEE FLOAT format:

Consider a temperature measurement of 36.4 Celsius with precision of 0.1 Celsius. The IEEE FLOAT representation is a 32-bit value consisting of an exponent of an 8-bit signed integer followed by a mantissa of a 24-bit signed integer; here, the exponent is -1 (0xFF) and the mantissa is 364 (0x00016C). Therefore, the IEEE FLOAT representation of 36.4 is 0xFF00016C.

2.2.2 SHORT FLOATING POINT TYPE (SFLOAT-TYPE) DATA TYPE

The following information is defined in ISO/IEEE Std. 11073-2060™1-2008 [1].

The SFLOAT-Type data type is defined to represent numeric values that are not integer in type. The SFLOAT-Type is defined as a 16-bit value with 12-bit mantissa and 4-bit exponent. See Annex F.8 of [1] for a thorough definition of the SFLOAT-Type. This data type is defined as follows:

	Exponent	**Mantissa**
Size	4 bit	12 bit

--
-- 16-bit float type; the integer type is a placeholder only
--
SFLOAT-Type ::= INT-U16

The 16–bit value contains a 4-bit exponent to base 10, followed by a 12-bit mantissa. Each is in twos-complement form.

Special values are assigned to express the following:

- NaN [exponent 0, mantissa +(2^11 –1) → 0x07FF]
- NRes [exponent 0, mantissa –(2^11) → 0x0800]
- + INFINITY [exponent 0, mantissa +(2^11 –2) → 0x07FE]
- – INFINITY [exponent 0, mantissa –(2^11 –2) → 0x0802]
- Reserved for future use [exponent 0, mantissa –(2^11 –1) → 0x0801]

Example

Example for blood pressure measurement in IEEE SFLOAT format:

Consider a systolic blood pressure measurement of 114 mmHg with a precision of 1 mmHg. The IEEE SFLOAT representation is a 16-bit value consisting of an exponent of a 4-bit signed integer followed by a mantissa of a12-bit signed integer; here, the exponent is 0 (0x0) and the mantissa is 114 (0x072). Therefore, the IEEE SFLOAT representation of 114 is 0x0072.

2.2.3 STRING CONVERSION

IEEE variable-length string type is encoded with a field length of 2 octets followed by the specific OCTET STRING data array. IEEE strings must be even length (16-bit aligned). For optimized data exchange over *Bluetooth* LE, no such requirement exists. *Bluetooth* characteristic strings can be odd or even length and the length of the string can be deciphered from the data. To transcode an odd length string, append a zero (0x00) byte to the end of the string, and increment the string length field.

Bluetooth characteristic strings are encoded as UTF-8, whereas IEEE strings are encoded as ASCII printable characters (a UTF-8 subset). Therefore, all implementations that aim to be IEEE compatible are required to restrict all their string characteristics to the ASCII printable subset.

2.2.4 MDS ATTRIBUTE CONVERSION

System-Model

The System-Model IEEE attribute consists of a sequence that contains manufacturer name and model number, respectively. Its content is vendor-decided, and represented as an OCTET STRING. Therefore, it must follow the string conversion rules as described in Section 2.2.3.

All fields of the System-Model attribute are derived from the characteristics in the DIS [2]. The *Bluetooth* “Manufacturer Name String” and “Model Number String” characteristics of the DIS map to Manufacturer Name and Model Number field of the System Model IEEE Attribute.

System-Id

The System-Id IEEE attribute has the same constraints as defined by the “System ID” *Bluetooth* characteristic (an EUI-64, which consists of a 24-bit Organizationally Unique Identifier followed by a 40-bit manufacturer-defined identifier). It is mapped directly from the DIS “System ID“ *Bluetooth* characteristic value. For more information, see the DIS [2].

Production-Specification

The IEEE attribute consists of a ProdSpecEntry list. Each entry may describe specific information such as serial number, hardware revision, software revision, protocol revision, firmware revision, and part numbers. Additionally, each entry in the list contains a Spec Type defining which type of specification it refers to, a vendor-specified component ID, and a vendor-specified ASCII printable string, mapped directly from the DIS characteristics as follows.

ProdSpecEntry Spec Type	*Bluetooth* Characteristic	IEEE Spec Type Value
Unspecified	N/A	0x0000
Serial Number	Serial Number String (DIS)	0x0001
Part Number	N/A	0x0002
Hardware Revision	Hardware Revision String (DIS)	0x0003
Software Revision	Software Revision String (DIS)	0x0004
Firmware Revision	Firmware Revision String (DIS)	0x0005
Protocol Revision	N/A	0x0006
GMDN (Global Medical Device Nomenclature)	N/A	0x0007

Table 2: Production Specification

The conversion is done as follows for each *Bluetooth* characteristic (Serial Number String, Hardware Revision String, Software Revision String, and Firmware Revision String):

1. Create a new ProdSpecEntry.
2. Set the first field (spec_type) according to table above.
3. Set the second field to the vendor-specified component ID.
4. Set the third field to the corresponding *Bluetooth* characteristic value.

Date-and-Time

See Section 2.2.6.

2.2.5 IEEE 11073-20601 REGULATORY CERTIFICATION DATA LIST

Health and Medical Devices may claim adherence to various regulatory and/or certification compliance items as an informative statement.

The IEEE 11073-20601 Regulatory Certification Data List enables a device to list the compliance items identifying the authorizing body and its data. IEEE, Continua, and the FDA are a few examples of authorizing bodies (although strictly speaking, IEEE is a Standards Development Organization).

The IEEE 11073-20601 Regulatory Certification Data List is defined as an opaque structure in the DIS by a regulatory body. The endianness of this data structure is as defined in the associated regulatory specification. For example, if the regulatory body is Continua, the “RegCertDataList” data structure will be in big-endian format.

Following is an example of this opaque structure based on Continua Design Guidelines 2010 (v1.5)[7].

In this example, the *IEEE 11073-20601 Regulatory Certification Data List* characteristic is required to have the following format:

Field Name	Offset	Size	Data Type	Definition / Notes
Regulatory Certification Data List	0			
Count	0	2 octet	uint16	
Length	2	2 octet	uint16	
Authorization Body	4	1 octet	uint8	Code assigned by IEEE identifying the authorizing body
Authorization Body Structure Type	5	1 octet	uint8	Identifies the data structure
Authorization Body Structure Length	6	2 octet	uint16	Defines authorization body data length
Authorizing Body Data	8	variable length	Opaque structure	Format defined by Authorizing Body
Major IG version	8	1 octet	uint8	
Minor IG version	9	1 octet	uint8	
Certified device class list	10			
Count	10	2 octet	uint16	
Length	12	2 octet	uint16	
Certified device class entry	12+n*2	2 octet	uint16	May be several of these entries, where n is the index of the device entry [1..*]
Continua Regulatory Structure	14+n*2	2 octet	uint16	
Structure length	16+n*2	2 octet	uint16	
Regulation Bit Field Type	18+n*2	2 octet	bits-16	

Table 3: Format Example for IEEE 11073-20601 Regulatory Certification Data List Characteristic

Table 4 is an example showing the contents of this opaque structure based on Continua Design Guidelines 2010 (v1.5)[7]. The minor and major Interoperability Guidelines (IG) version represents the current Continua Guideline being followed, and will be updated as new guidelines are adopted and followed by this document.

In this example, the Continua Health Alliance is the regulatory body and the device includes only one device specialization – the IEEE 11073-10408 Thermometer device specialization [5]. As a result, the *IEEE 11073-20601 Regulatory Certification Data List* characteristic is required to have the following format and values:

Data	Description
0x00 0x02	RegCertDataList.count = 2
0x00 0x12	RegCertDataList.length = 18
0x02 0x01	RegCertDataList[0].auth-body-and-struc-type.: auth-body = auth-body-continua = 2 auth-body-struc-type = continua-version-struct = 1 (*ContinuaBodyStruct*)
0x00 0x08	RegCertDataList[0]. auth-body-data.length = 8
0x01 0x05	RegCertDataList[0]. auth-body-data: ContinuaBodyStruct.major-IG-version = 1 ContinuaBodyStruct.minor-IG-version = 5
0x00 0x01	CertifiedDeviceClassList.count = 1
0x00 0x02	CertifiedDeviceClassList.length = 2
0x40 0x08	CertifiedDeviceClassList. CertifiedDeviceClassEntry = 0x4008 Based on Continua 2010 (v1.5) guidelines for a Low Power Wireless PAN Thermometer:Transport Code (TCode) = 4 (Low Power Wireless PAN) MDC_DEV_SPEC_PROFILE_TEMP = 0x10 0x08 = 4104_{10} CertifiedDeviceClass = 4104-4096+4*8192 = 32776_{10} -> 0x8008
0x02 0x02	RegCertDataList[1].auth-body-and-struct-type: auth-body = auth-body-continua = 2 auth-body-struc-type = continua-reg-struct = 2 (*ContinuaRegStruct*)
0x00 0x02	RegCertDataList[0]. auth-body-data.length = 2
0x00 0x00	This is a regulated device

Table 4: IEEE 11073-20601 Regulatory Certification Data List Characteristic Example

In this example, the total length of the structure is 22 octets.

2.2.6 TRANSCODING TIME STAMP TO IEEE ABSOLUTETIME

For the *Bluetooth* profiles encompassed in this white paper, the measurement time stamp follows the format of the *Bluetooth* Date Time characteristic. This format is encoded using a hexadecimal format, according to the *Bluetooth* Assigned Numbers [4]. Each field has 8 bits, except the first one that encodes the year, which is 16 bits. For example, the time stamp for 18th December 2010, 15:23:06 is encoded as 0x07DA 0x0C 0x12 0x0F 0x17 0x06. IEEE AbsoluteTime data format is also used to define date and time, but it is encoded using binary coded decimal (i.e., 4-bit nibbles) and every field has 8 bits. For example, the time stamp for 18th December 2010, 15:23:06:73 is encoded as 0x20 0x10 0x12 0x18 0x15 0x23 0x06 0x73. IEEE AbsoluteTime specifies time with a resolution of 1/100 of a second and the

Bluetooth Date Time characteristic has resolution of 1 second, so the Centiseconds field must be set to zero in the transcoding process.

As AbsoluteTime is encoded as a Binary Coded Decimal (BCD) format, a conversion between formats is needed; however this conversion does not result in a loss of precision.

Table 5 shows mapping from the *Bluetooth* Time Stamp fields to the IEEE AbsoluteTime fields:

***Bluetooth* Field Name (Date Time)**	**IEEE Field Name (AbsoluteTime)**
Year[1]	century
	year
Month	month
Day	day
Hours	hour
Minutes	minute
Seconds	second
N/A[2]	sec-fractions

Table 5: Bluetooth Characteristic to IEEE Time Conversion

Notes:

1. The Year field of the *Bluetooth* Date Time characteristic is equivalent to the century and year IEEE attributes combined together. This conversion can be done by first converting the *Bluetooth* characteristic Year to BCD. The first octet represents the IEEE Century field, and the second octet represents the IEEE Year field.
2. It is envisioned that one-second precision for *Bluetooth* LE devices is sufficient. Fractions of seconds are not used in order to optimize the exchange of data over the air.

3. Device Specific Data Requirements

This section describes the mapping of device specific data from a *Bluetooth* environment to an IEEE environment. This section will be expanded for various *Bluetooth* profiles and services in the future as they become available.

3.1 HEALTH THERMOMETER

This section defines transcoding the thermometer device specific data into IEEE 11073-10408 Thermometer device specialization [5] class attributes.

3.1.1 DEVICE-SPECIFIC MDS CLASS REQUIREMENTS

In addition to the MDS class requirements shown in Section 2.1, Table 6 shows incremental MDS class requirements specific to this device.

IEEE attribute	*Bluetooth* equivalent characteristic	*Bluetooth* Service	*Bluetooth* data type	IEEE attribute type (ASN.1)	IEEE data type (informative)
System-Type	None	N/A	N/A	TYPE[1]	(INT-U16, INT-U16)
Dev-Configuration-Id	None	N/A	N/A	ConfigId[2]	INT-U16
System-Type-Spec-List	None	N/A	N/A	TypeVerList[3]	List of (INT-U16, INT-U16)
Confirm-Timeout	None	N/A	N/A	RelativeTime	OCTET STRING (SIZE(6))

Table 6: Device-specific MDS Class Requirements

Notes:

1. Value not present since System-Type-Spec-List exists.
2. Set to any value in range of 0x4000 to 0x7FFF (Extended Configuration). This is to assure that the Manager requests the configuration and does not assume it, as it could for a Standard Configuration.
3. Since the “Health Thermometer Service“ [6] is a “Primary Service“, an entry is required to be added to the TypeVerList as follows:

   0x10 0x08 type = MDC_DEV_SPEC_PROFILE_TEMP
   0x00 0x01 version = version 1 of the IEEE device specialization

3.1.2 IEEE NUMERIC CLASS REQUIREMENTS

This section describes the IEEE numeric class requirements. It is restricted to those IEEE attributes that are used on the Health Thermometer device. All unmentioned attributes defined in [1] are not applicable to a Health Thermometer device.

IEEE attribute	*Bluetooth* equivalent characteristic	*Bluetooth* Service	*Bluetooth* data type	IEEE attribute type (ASN.1)	IEEE data type (informative)
Handle	None	N/A	N/A	HANDLE[1]	INT-U16
Type	None	N/A	N/A	TYPE[2]	(INT-U16, INT-U16)
Metric-Spec-Small	None	N/A	N/A	MetricSpecSmall[3]	BITS-16
Unit-Code	Temperature Measurement[4]	Health Thermometer	Opaque Structure	OID-Type	INT-U16
Attribute-Value-Map	None	N/A	N/A	AttrValMap[7]	List of (INT-U16, INT-U16)
Absolute-Time-Stamp	Temperature Measurement[6]	Health Thermometer	Opaque Structure	AbsoluteTime	(INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8)
Simple-Nu-Observed-Value	Temperature Measurement[7]	Health Thermometer	Opaque Structure	SimpleNuObsValue	FLOAT-Type

Table 7: IEEE Numeric Class Requirements

Notes:

1. Each object is required to have a unique non-zero ID assigned by the implementation.
2. The value can be inferred based on the Temperature Type value. Although [1] indicates that supplemental information covers the location of the sensor, the thermometer device specialization [5], defines that an Extended Configuration with type {MDC_PART_SCADA, MDC_TEMP_zzz} is required to be used to report a specific temperature type.

IEEE Temperature type code	*Bluetooth* Temperature type Description	*Bluetooth* Value
MDC_TEMP_AXILLA	Armpit	0x01
MDC_TEMP_BODY	Body (general)	0x02
MDC_TEMP_EAR	Ear (usually ear lobe)	0x03
MDC_TEMP_FINGER	Finger	0x04
MDC_TEMP_GIT	Gastro-intestinal Tract	0x05
MDC_TEMP_ORAL	Mouth	0x06
MDC_TEMP_RECT	Rectum	0x07
MDC_TEMP_TOE	Toe	0x08
MDC_TEMP_TYMP	Tympanum (ear drum)	0x09
	Reserved	All other values

Table 8: Temperature Type Description Conversion

3. When the Measurement Interval characteristic is not present or when it is present and its value is zero (aperiodic mode) then this is set to 0xF040 (mss-avail-intermittent, mss-avail-stored-data, mss-upd-aperiodic, mss-msmt-aperiodic, mss- acc-agent-initiated). When the Measurement Interval characteristic is present and its value is non-zero (periodic mode) this is set to 0x4040 (mss-avail-stored-data, mss- acc-agent-initiated).

4. This value is mapped from Bit 0 of the least significant octet of the Temperature Measurement characteristic. The mapping is as follows in Table 9:

IEEE Temperature Unit Value	*Bluetooth* Temperature Unit Value	Temperature Unit description
MDC_DIM_DEGC	0	Celsius
MDC_DIM_FAHR	1	Fahrenheit

Table 9: Temperature Type Description Conversion

5. This IEEE attribute is transmission-related only. As this document is only relevant for nomenclature and model compatibility and does not mandate any way to reach the IEEE domain, this attribute is implementation specific.
6. When supported, this value is extracted from the Time Stamp field of Temperature Measurement characteristic. See Section 2.2.6.
7. This value is extracted from the Temperature Measurement value of the Temperature Measurement characteristic.

4. End-To-End Example

This section provides an example of an end-to-end communication between a *Bluetooth* low energy Health Thermometer and a Collector (e.g. phone) implementing an IEEE Manager and a transcoder. It describes how *Bluetooth* characteristic data can be mapped to IEEE nomenclature and modeling. This example illustrates the steps taken to use the mappings so that the transcoder would be capable of generating IEEE APDUs based on the received data.

Hypothetical Health Thermometer data is used in Section 4.1 as an input. Section 4.3 discusses how this data could be mapped into IEEE objects.

4.1 HEALTH THERMOMETER DATA

Table 10 and Table 11 describe the Health Thermometer data being sent to the Collector which implements a Transcoder. The Health Thermometer *Bluetooth* Address is 00:23:6C:AF:BD:F4.

Health Thermometer Service Data

This data refers to a previous time stamped measurement taken from a general body location.

Bluetooth **Characteristic**	***Bluetooth*** **Value**
Date Time	18th December 2010 15:23:06 encoded as: 0x07DA 0x0C 0x12 0x0F 0x17 0x06
Temperature Measurement	37.0°C with Time Stamp of 18th December 2010 15:00:00: 0xFF 0x00 0x01 0x72 0x07 0xDA 0x0C 0x12 0x0F 0x00 0x00
Temperature Type	Body: 0x02
Intermediate Temperature	Not transcoded to IEEE.
Measurement Interval	Not transcoded to IEEE.
Valid Range Descriptor	Not transcoded to IEEE.

Table 10: Health Thermometer Service Data

Device Information Service Data

This data refers to a Health Thermometer that has its System ID filled in based on its *Bluetooth* address, as described in the Bluetooth SIG Assigned Numbers [4].

Bluetooth **Characteristic**	***Bluetooth*** **Value**
System ID	0x00 0x23 0x6C 0xFF 0xFE 0xAF 0xBD 0xF4
Model Number String	“TS-1017”
Manufacturer Name String	“ACME”
Serial Number String	“237495-3282-A”
Firmware Revision String	“1.23”
Hardware Revision String	“1.0”
Software Revision String	“1.2”
IEEE 11073-20601 Regulatory Certification Data List	For an example on how to populate this structure, refer to Section 2.2.5.

Table 11: Device Information Service data

4.2 HEALTH THERMOMETER SERVICE RECORD

Table 12 shows the Health Thermometer Service record and the attributes contained on the server.

***Bluetooth* Attribute**	***Bluetooth* Attribute Value**	**Description**
Primary Service (0x2800)	0x180A	Device Information Service
Characteristic (0x2803)	{0x02, 0xhhhh, 0x2A23}	Characteristic Value is Read, Value Handle reference is 0xhhhh and Characteristic is "System ID"
System ID (0x2A23)	{0x00236CFFFEAFBDF4}	System ID
Characteristic (0x2803)	{0x02, 0xhhhh, 0x2A29}	Characteristic Value is Read, Value Handle reference is 0xhhhh and Characteristic is "Manufacturer Name String"
Manufacturer Name String (0x2A29)	{0x41, 0x43, 0x4D, 0x45}	Manufacturer Name String = UTF8 String "ACME"
Characteristic (0x2803)	{0x02, 0xhhhh, 0x2A24}	Characteristic Value is Read, Value Handle reference is 0xhhhh and Characteristic is "Model Number String"
Model Number String (0x2A24)	{0x54, 0x53, 0x2D, 0x31, 0x30, 0x31, 0x37}	Model Number String = UTF8 String "TS-1017"
Characteristic (0x2803)	{0x02, 0xhhhh, 0x2A25}	Characteristic Value is Read, Value Handle reference is 0xhhhh and Characteristic is "Serial Number String"
Serial Number String (0x2A25)	{0x32, 0x33, 0x37, 0x34, 0x39, 0x35, 0x2D, 0x33, 0x32, 0x38, 0x32, 0x2D, 0x41}	Serial Number String = UTF8 String "237495-3282-A"
Characteristic (0x2803)	{0x02, 0xhhhh, 0x2A26}	Characteristic Value is Read, Value Handle reference is 0xhhhh and Characteristic is "Firmware Revision String"
Firmware Revision String (0x2A26)	{0x31, 0x2E, 0x32, 0x33}	Firmware Revision String = UTF8 String "1.23"
Characteristic (0x2803)	{0x02, 0xhhhh, 0x2A27}	Characteristic Value is Read, Value Handle reference is 0xhhhh and Characteristic is "Hardware Revision"
Hardware Revision String (0x2A27)	{0x31, 0x2E, 0x30}	Hardware Revision String = UTF8 String "1.0"
Characteristic (0x2803)	{0x02, 0xhhhh, 0x2A28}	Characteristic Value is Read, Value Handle reference is 0xhhhh and Characteristic is "Software Revision String"
Serial Number String (0x2A28)	{0x31, 0x2E, 0x32}	Software Revision String = UTF8 String "1.2"
Primary Service (0x2800)	0x1809	Health Thermometer Service
Characteristic (0x2803)	0x20, 0xhhhh, 0x2A1C	Characteristic Value is Indicated, Value Handle reference is 0xhhhh and Characteristic is "Temperature Measurement"
Temperature Measurement (0x2A1C)	{0x02, 0xFF000172, 0x07DA 0x0C 0x12 0x0F 0x17 0x06}	Timestamp [18th December 2010 15:23:06] Temperature Measurement of 37.0 degree in Celsius

Table 12: Health Thermometer Service Record

4.3 IEEE OBJECTS

The following tables describe how data are represented as IEEE objects.

MDS Object

IEEE attribute	*Bluetooth* equivalent characteristic	IEEE Value	Reference
Handle	None	0	
System-Model	Model Number String	"TS-1017" [1]	Section 2.2.4 – System Model
	Manufacturer Name String	"ACME"	
System-Id	System ID	0x00 0x23 0x6C 0xFF 0xFE 0xAF 0xBD 0xF4	Section 2.2.4 – System Id
Dev-Configuration-Id	None	MDC_TEMP_BODY = 0x4002	Section 3.1.1
Production-Specification	Serial Number String	"237495-3282-A" [1]	Section 2.2.4 – Prod. Specification
	Hardware Revision String	"1.0" [1]	
	Software Revision String	"1.2" [1]	
	Firmware Revision String	"1.23"	
Date-and-Time	Date Time	0x20 0x10 0x0C 0x12 0x0F 0x17 0x06 0x00	Section 2.2.6
Reg-Cert-Data-List	IEEE 11073-20601 Regulatory Certification Data List	For an example on how to populate this structure, refer to Section 2.2.5.	Section 2.2.4 – Reg-Cert-Data-List
System-Type-Spec-List	None	{MDC_DEV_SPEC_PROFILE_TEMP, 1}	

Table 13: MDS Object for Health Thermometer

Notes:

1. Because this is an uneven string, a zero (0x00) byte must be appended to its end and its length field must be incremented. See Section 2.2.3 for more information.

Numeric Object

IEEE attribute	*Bluetooth* equivalent characteristic	Value	Reference
Handle	None	1	
Type	None	MDC_PART_SCADA	Section 3.1.1
	None	MDC_TEMP_BODY	
Metric-Spec-Small	None		
Measurement-Status	None	relevant data, 0x10	
Metric-Id	None	MDC_TEMP_BODY = 0x4002	Section 3.1.2
Metric-Id-List	None		
Metric-Id-Partition	None		
Unit-Code	Temperature Measurement	MDC_DIM_DEGC	Section 3.1.2
Attribute-Value-Map	None		
Absolute-Time-Stamp	Temperature Measurement	0x20 0x10 0x0C 0x12 0x0F 0x00 0x00	Section 3.1.2
Measure-Active-Period	None		
Simple-Nu-Observed-Value	None	0xFF000172 (37.0)	Section 3.1.2 and Annex F.8 of [1]
Compound-Simple-Nu-Observed-Value	None		
Basic-Nu-Observed-Value	None		
Compound-Basic-Nu-Observed-Value	None		
Nu-Observed-Value	Temperature Measurement		
Compound-Nu-Observed-Value	None		
Accuracy	None		

Table 14: Numeric Object for Health Thermometer

5. Acronyms and Abbreviations

Acronyms and Abbreviations	Meaning
APDU	Application Protocol Data Unit
ASCII	American Standard Code for Information Interchange as defined in ISO/IEC 646 (1991)
ASN.1	Abstract Syntax Notation One
BCD	Binary-Coded Decimal
DIS	Device Information Service
EUI	Extended Unique Identifier
FDA	Food and Drug Administration
GATT	Generic Attribute Profile
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
LE	Low Energy
MDS	Medical Device System
USB	Universal Serial Bus
UTF-8	Unicode Transformation Format-8
UUID	Universally Unique Identifier

Table 15: Acronyms and Abbreviations

6. References

[1] ISO/IEEE Std 11073-20601™- 2008 Health Informatics - Personal Health Device Communication - Application Profile - Optimized Exchange Protocol - version 1.0.This also includes ISO/IEEE Std 11073-20601a™-2010 – Amendment 1

[2] Device Information Service d10r05

[3] Health Thermometer Profile d10r07

[4] Bluetooth SIG Assigned Numbers

[5] Health informatics - Personal health device communication - Part 10408: Device specialization - Thermometer

[6] Health Thermometer Service d10r06

[7] Continua Design Guidelines 2010 (version 1.5)