4/6/2016

Bluetooth Developer Studio Level 2 Profile Report

PROFILE

Profile Name

BBC MICROBIT

Abstract:

Default 'out of the box' profile for the BBC Micro Bit

Summary:

```
Version 1.7 - 22nd January 2016

Standard Bluetooth pairing and security are now used. Specifically:
1. Pairing with passkey and MITM protection
2. White Listing
3. Encrypted link for most operations

All services except Generic Access, Generic Attribute, Device Information and DFU Control Service designated OPTIONAL
DFU Control Service has lost the the DFU Flash Code characteristic since we're now using standard Bluetooth pairing.
Changed names of button characteristics to use A and B instead of 1 and 2
Revised 5 byte representation of the LED Matrix:
 Octet 0, LED Row 1: bit4 bit3 bit2 bit1 bit0
 Octet 1, LED Row 2: bit4 bit3 bit2 bit1 bit0
 Octet 2, LED Row 3: bit4 bit3 bit2 bit1 bit0
 Octet 3, LED Row 4: bit4 bit3 bit2 bit1 bit0
 Octet 4, LED Row 5: bit4 bit3 bit2 bit1 bit0
Maximum length of LED Text documented.
Changed name of "Scrolling Speed" characteristic to "Scrolling Delay".
Reinstated Manufacturer Name String characteristic to the Device Information Service.
DFU Control characteristic given the READ property
Documented supported values the accelerometer and magnetometer period characteristics can take.
Documented magic event type/value of zero
Documented event type/value are little endian

Version 1.6 - 17th October 2015
Removed the Battery Service. No way to establish battery levels on the micro:bit
Added a simple Temperature Service to exploit temperature sensors in micro:bit processors with Temperature and Temperature Period characte
Accelerometer and Magnetometer period characteristics now have uint16 fields instead of uint8 which required scaling up by multipling by 1
Accelerometer Data and Magnetometer Data characteristics now use signed 16 bit integer fields for each of their X, Y and Z parts.
Accelerometer Data and Magnetometer Data characteristics now use signed 16 bit integer fields for each of their X, Y and Z parts.
New characteristic Magnetometer Heading added to the Magnetometer Service. Provides current heading in degrees.
Removed IO Parallel Port characteristic due to complexity and memory considerations.
Added Generic Attribute Service (previously absent in the repository)
Changed the LED Matrix State characteristic field so that we now have one octet per row of LEDs for ease of use.

Version 1.5 - 10th September 2015
Button State 2 characteristic given new, distinct UUID of E95DDA91-251D-470A-A062-FA1922DFA9A8
Removed the System LED State characteristic from the LED Service since it cannot be controlled from the BLE MCU.
Removed the Scrolling State characteristic from the LED Service due to complexity and memory constraints.
Changed LED Matrix State use of “Write Without Response” to “Write” so that no further writes can be made until there’s been an ACK back f
Removed Write property from MicroBit Requirements characteristic.
```

Base UUID	E95D0000251D470AA062FA1922DFA9A8
Server Role	
Client Role	

SERVICES

Generic Access

UUID	00001800000010008000000805F9B34FB
Declaration	Primary
Requirement	Mandatory
Server Role	
Client Role	

Abstract:

The generic_access service contains generic information about the device. All available Characteristics are readonly.

Summary:

Examples:

Generic Access - CHARACTERISTICS

Device Name

UUID	00002A0000001000800000805F9B34FB
Type	
Requirement	Mandatory
Abstract:	
Summary:	
Examples	
Read	Mandatory
Write	Mandatory
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	

Appearance

UUID	00002A0100001000800000805F9B34FB
Type	
Requirement	Mandatory
Abstract:	

The external appearance of this device. The values are composed of a category (10-bits) and sub-categories (6-bits).

Summary:	
Examples	
Read	Mandatory
Write	Excluded
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	

Peripheral Preferred Connection Parameters

UUID	00002A0400001000800000805F9B34FB
Type	
Requirement	Mandatory
Abstract:	

Summary:	
Examples	
Read	Mandatory
Write	Excluded
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	

Generic Attribute

UUID	00001801000010008000000805F9B34FB
Declaration	Primary
Requirement	Mandatory
Server Role	
Client Role	
Abstract:	
Summary:	
Examples:	

Generic Attribute - CHARACTERISTICS

Service Changed

UUID	2A05
Type	
Requirement	Optional
Abstract:	
Summary:	
Examples	
Read	Excluded
Write	Excluded
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Mandatory
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	1. Client Characteristic Configuration : 2902

Device Information

UUID	0000180A000010008000000805F9B34FB
Declaration	Primary
Requirement	Mandatory
Server Role	
Client Role	
Abstract:	

The Device Information Service exposes manufacturer and/or vendor information about a device.

Summary:

This service exposes manufacturer information about a device.
The Device Information Service is instantiated as a Primary Service.
Only one instance of the Device Information Service is exposed on a device.

Examples:

Device Information - CHARACTERISTICS

Model Number String

UUID	00002A2400001000800000805F9B34FB
Type	
Requirement	Optional
Abstract:	

The value of this characteristic is a UTF-8 string representing the model number assigned by the device vendor.

Summary:

Examples

Read	Mandatory
Write	Excluded
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded

Descriptors

Serial Number String

UUID	00002A2500001000800000805F9B34FB
Type	
Requirement	Optional
Abstract:	

The value of this characteristic is a variable-length UTF-8 string representing the serial number for a particular instance of the dev

Summary:

Examples

Read	Mandatory
Write	Excluded
Write Without Response	Excluded

Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	

Hardware Revision String

UUID	00002A2700001000800000805F9B34FB
Type	
Requirement	Optional
Abstract:	
Summary:	

The value of this characteristic is a UTF-8 string representing the hardware revision for the hardware within the device.

Examples	
Read	**Mandatory**
Write	Excluded
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	

Firmware Revision String

UUID	00002A2600001000800000805F9B34FB
Type	
Requirement	Optional
Abstract:	
Summary:	

The value of this characteristic is a UTF-8 string representing the firmware revision for the firmware within the device.

Examples	
Read	**Mandatory**
Write	Excluded
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded

Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	

Manufacturer Name String

UUID	00002A2900001000800000805F9B34FB
Type	
Requirement	Mandatory
Abstract:	

The value of this characteristic is a UTF-8 string representing the name of the manufacturer of the device.

Summary:	
Examples	
Read	Mandatory
Write	Excluded
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	

ACCELEROMETER SERVICE

UUID	E95D0753251D470AA062FA1922DFA9A8
Declaration	Primary
Requirement	Optional
Server Role	
Client Role	
Abstract:	
Summary:	

Exposes accelerometer data. An accelerometer is an electromechanical device that will measure acceleration forces.
These forces may be static, like the constant force of gravity pulling at your feet, or they could be dynamic - caused by moving or vibrat

Value contains fields which represent 3 seperate accelerometer measurements for X, Y and Z axes as 3 unsigned 16 bit values in that order
little endian format.

Data can be read on demand or notified periodically.

Examples:	

ACCELEROMETER SERVICE - CHARACTERISTICS

Accelerometer Data

UUID	E95DCA4B251D470AA062FA1922DFA9A8
Type	
Requirement	Mandatory
Abstract:	
Summary:	

Contains accelerometer measurements for X, Y and Z axes as 3 signed 16 bit values in that order and in little endian format.

Examples

Read	**Mandatory**
Write	Excluded
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	**Mandatory**
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	1. Client Characteristic Configuration : 2902

Accelerometer Period

UUID	E95DFB24251D470AA062FA1922DFA9A8
Type	
Requirement	**Mandatory**
Abstract:	

Summary:

Determines the frequency with which accelerometer data is reported in milliseconds.

Valid values are 1, 2, 5, 10, 20, 80, 160 and 640.

Examples

Read	**Mandatory**
Write	**Mandatory**
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	

MAGNETOMETER SERVICE

UUID	E95DF2D8251D470AA062FA1922DFA9A8
Declaration	Primary
Requirement	Optional
Server Role	
Client Role	
Abstract:	

Summary:

Exposes magnetometer data. A magnetometer measures a magnetic field such as the earth's magnetic field in 3 axes.

Examples:

MAGNETOMETER SERVICE - CHARACTERISTICS

Magnetometer Data

UUID	E95DFB11251D470AA062FA1922DFA9A8
Type	
Requirement	Mandatory
Abstract:	
Summary:	

Contains magnetometer measurements for X, Y and Z axes as 3 signed 16 bit values in that order and in little endian format. Data can be read on demand or notified periodically.

Examples	
Read	Mandatory
Write	Excluded
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Mandatory
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	1. Client Characteristic Configuration : 2902

Magnetometer Period

UUID	E95D386C251D470AA062FA1922DFA9A8
Type	
Requirement	Mandatory
Abstract:	
Summary:	

Determines the frequency with which magnetometer data is reported in milliseconds.

Valid values are 1, 2, 5, 10, 20, 80, 160 and 640.

Examples	
Read	Mandatory
Write	Mandatory
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	

Magnetometer Bearing

UUID	E95D9715251D470AA062FA1922DFA9A8

Type	
Requirement	Mandatory
Abstract:	
Summary:	
Compass bearing in degrees from North.	
Examples	
Read	Mandatory
Write	Excluded
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Mandatory
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	1. Client Characteristic Configuration : 2902

Button Service

UUID	E95D9882251D470AA062FA1922DFA9A8
Declaration	Primary
Requirement	Optional
Server Role	
Client Role	
Abstract:	
Summary:	
Exposes the two Micro Bit buttons and allows 'commands' associated with button state changes to be associated with button states and notif	
Examples:	

Button Service - CHARACTERISTICS

Button A State

UUID	E95DDA90251D470AA062FA1922DFA9A8
Type	
Requirement	Mandatory
Abstract:	
Summary:	
State of Button A may be read on demand by a connected client or the client may subscribe to notifications of state change. 3 button states are defined and represented by a simple numeric enumeration: 0 = not pressed, 1 = pressed, 2 = long press.	
Examples	
Read	Mandatory
Write	Excluded
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Mandatory

Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	1. Client Characteristic Configuration : 2902

Button B State

UUID	E95DDA91251D470AA062FA1922DFA9A8
Type	
Requirement	Mandatory
Abstract:	
Summary:	

State of Button B may be read on demand by a connected client or the client may subscribe to notifications of state change.
3 button states are defined and represented by a simple numeric enumeration: 0 = not pressed, 1 = pressed, 2 = long press.

Examples

Read	Mandatory
Write	Excluded
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Mandatory
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	1. Client Characteristic Configuration : 2902

IO PIN SERVICE

UUID	E95D127B251D470AA062FA1922DFA9A8
Declaration	Primary
Requirement	Optional
Server Role	
Client Role	
Abstract:	
Summary:	

Provides read/write access to I/O pins, individually or collectively. Allows configuration of each pin for input/output and analogue/digit

Examples:

IO PIN SERVICE - CHARACTERISTICS

Pin Data

UUID	E95D8D00251D470AA062FA1922DFA9A8
Type	
Requirement	Mandatory
Abstract:	
Summary:	

Contains data relating to zero or more pins. Structured as a variable length array of up to 19 Pin Number / Value pairs.

Pin Number and Value are each uint8 fields.

Note however that the micro:bit has a 10 bit ADC and so values are compressed to 8 bits with a loss of resolution.

OPERATIONS:

WRITE: Clients may write values to one or more pins in a single GATT write operation.
A pin to which a value is to be written must have been configured for output using the Pin IO Configuration characteristic.
Any attempt to write to a pin which is configured for input will be ignored.

NOTIFY: Notifications will deliver Pin Number / Value pairs for those pins defined as input pins by the Pin IO Configuration character
and whose value when read differs from the last read of the pin.

READ: A client reading this characteristic will receive Pin Number / Value pairs for all those pins defined as input pins by the Pin I

Examples

Read	**Mandatory**
Write	**Mandatory**
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	**Mandatory**
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	1. Client Characteristic Configuration : 2902

Pin AD Configuration

UUID	E95D5899251D470AA062FA1922DFA9A8
Type	
Requirement	**Mandatory**
Abstract:	

Summary:

A bit mask which allows each pin to be configured for analogue or digital use.

Bit n corresponds to pin n where 0 LESS THAN OR EQUAL TO n LESS THAN 19. A value of 0 means digital and 1 means analogue.

Examples

Read	**Mandatory**
Write	**Mandatory**
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	

Pin IO Configuration

UUID	E95DB9FE251D470AA062FA1922DFA9A8
Type	
Requirement	**Mandatory**

Abstract:	
Summary:	

A bit mask which allows each pin to be configured for input or output use.

Bit n corresponds to pin n where 0 LESS THAN OR EQUAL TO n LESS THAN 19. A value of 0 means configured for output and 1 means configur

Examples	
Read	Mandatory
Write	Mandatory
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	

LED SERVICE

UUID	E95DD91D251D470AA062FA1922DFA9A8
Declaration	Primary
Requirement	Optional
Server Role	
Client Role	
Abstract:	
Summary:	

Provides access to and control of LED state. Allows the state (ON or OFF) of all 25 LEDs to be set in a single write operation.
Allows short text strings to be sent by a client for display on the LED matrix and scrolled across at a speed controlled by the Scrolling

Examples:

LED SERVICE - CHARACTERISTICS

LED Matrix State

UUID	E95D7B77251D470AA062FA1922DFA9A8
Type	
Requirement	Mandatory
Abstract:	
Summary:	

Allows the state of any|all LEDs in the 5x5 grid to be set to on or off with a single GATT operation.
Consists of an array of 5 x utf8 octets, each representing one row of 5 LEDs.
Octet 0 represents the first row of LEDs i.e. the top row when the micro:bit is viewed with the edge connector at the bottom and USB c
Octet 1 represents the second row and so on.
In each octet, bit 4 corresponds to the first LED in the row, bit 3 the second and so on.
Bit values represent the state of the related LED: off (0) or on (1).

So we have:

Octet 0, LED Row 1: bit4 bit3 bit2 bit1 bit0
Octet 1, LED Row 2: bit4 bit3 bit2 bit1 bit0
Octet 2, LED Row 3: bit4 bit3 bit2 bit1 bit0
Octet 3, LED Row 4: bit4 bit3 bit2 bit1 bit0
Octet 4, LED Row 5: bit4 bit3 bit2 bit1 bit0

Examples	
Read	Mandatory

Write	**Mandatory**
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	

LED Text

UUID	E95D93EE251D470AA062FA1922DFA9A8
Type	
Requirement	**Mandatory**
Abstract:	
Summary:	

A short UTF-8 string to be shown on the LED display. Maximum length 20 octets.

Examples	
Read	Excluded
Write	**Mandatory**
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	

Scrolling Delay

UUID	E95D0D2D251D470AA062FA1922DFA9A8
Type	
Requirement	**Mandatory**
Abstract:	
Summary:	

Specifies a millisecond delay to wait for in between showing each character on the display.

Examples	
Read	**Mandatory**
Write	**Mandatory**
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded

Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	

EVENT SERVICE

UUID	E95D93AF251D470AA062FA1922DFA9A8
Declaration	Primary
Requirement	Optional
Server Role	
Client Role	
Abstract:	

Summary:

A generic, bi-directional event communication service.

The Event Service allows events or commands to be notified to the micro:bit by a connected client and it allows micro:bit to notify the co
of events or commands originating from with the micro:bit. The micro:bit can inform the client of the types of event it is interested in b
about (e.g. an incoming call) and the client can inform the micro:bit of types of event it wants to be notified about.

The term “event” will be used here for both event and command types of data.

Events may have an associated value.

Note that specific event ID values including any special values such as those which may represent wild cards are not defined here.
The micro:bit run time documentation should be consulted for this information.

Multiple events of different types may be notified to the client or micro:bit at the same time.
Event data is encoded as an array of structs each encoding an event of a given type together with an associated value.
Event Type and Event Value are both defined as uint16 and therefore the length of this array will always be a multiple of 4.

```
struct event {
 uint16 event_type;
 uint16 event_value;
};
```

Examples:

EVENT SERVICE - CHARACTERISTICS

MicroBit Requirements

UUID	E95DB84C251D470AA062FA1922DFA9A8
Type	
Requirement	Mandatory
Abstract:	

Summary:

A variable length list of event data structures which indicates the types of client event, potentially with a specific value which the
to be informed of when they occur. The client should read this characteristic when it first connects to the micro:bit. It may also sub
to that it can be informed if the value of this characteristic is changed by the micro:bit firmware.

```
struct event {
  uint16 event_type;
  uint16 event_value;
};
```

Note that an event_type of zero means ANY event type and an event_value part set to zero means ANY event value.

event_type and event_value are each encoded in little endian format.

Examples

Read	Mandatory
Write	Excluded
Write Without Response	Excluded

Signed Write	Excluded
Reliable Write	Excluded
Notify	Mandatory
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	1. Client Characteristic Configuration : 2902

MicroBit Event

UUID	E95D9775251D470AA062FA1922DFA9A8
Type	
Requirement	Mandatory
Abstract:	

Summary:

Contains one or more event structures which should be notified to the client. It supports notifications and as such the client should notifications from this characteristic.

```
struct event {
  uint16 event_type;
  uint16 event_value;
};
```

Examples

Read	Mandatory
Write	Excluded
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Mandatory
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	1. Client Characteristic Configuration : 2902

Client Requirements

UUID	E95D23C4251D470AA062FA1922DFA9A8
Type	
Requirement	Mandatory
Abstract:	

Summary:

a variable length list of event data structures which indicates the types of micro:bit event, potentially with a specific value which to be informed of when they occur. The client should write to this characteristic when it first connects to the micro:bit.

```
struct event {
  uint16 event_type;
  uint16 event_value;
};
```

Note that an event_type of zero means ANY event type and an event_value part set to zero means ANY event value.

event_type and event_value are each encoded in little endian format.

Examples	
Read	Excluded
Write	Mandatory
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	

Client Event

UUID	E95D5404251D470AA062FA1922DFA9A8
Type	
Requirement	Mandatory
Abstract:	
Summary:	

a writable characteristic which the client may write one or more event structures to, to inform the micro:bit of events which have occ
These should be of types indicated in the micro:bit Requirements characteristic bit mask.

```
struct event {
  uint16 event_type;
  uint16 event_value;
};
```

Examples	
Read	Excluded
Write	Mandatory
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	

DFU CONTROL SERVICE

UUID	E95D93B0251D470AA062FA1922DFA9A8
Declaration	Primary
Requirement	Mandatory
Server Role	
Client Role	
Abstract:	
Summary:	

Allows clients to initiate the micro:bit pairing and over the air firmware update procedures.

Examples:

DFU CONTROL SERVICE - CHARACTERISTICS

DFU Control

UUID	E95D93B1251D470AA062FA1922DFA9A8
Type	
Requirement	Mandatory
Abstract:	

Summary:

Writing 0x01 initiates rebooting the micro:bit into the Nordic Semiconductor bootloader if the DFU Flash Code characteristic has been to with the correct secret key.

Writing 0x02 to this characteristic means "request flash code".

Examples

Read	Mandatory
Write	Mandatory
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	

TEMPERATURE SERVICE

UUID	E95D6100251D470AA062FA1922DFA9A8
Declaration	Primary
Requirement	Optional
Server Role	
Client Role	
Abstract:	

Summary:

Ambient temperature derived from several internal temperature sensors on the micro:bit

Examples:

TEMPERATURE SERVICE - CHARACTERISTICS

Temperature

UUID	E95D9250251D470AA062FA1922DFA9A8
Type	
Requirement	Mandatory
Abstract:	

Summary:

Signed integer 8 bit value in degrees celsius.

Examples

Read	**Mandatory**
Write	Excluded
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	**Mandatory**
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	1. Client Characteristic Configuration : 2902

Temperature Period

UUID	E95D1B25251D470AA062FA1922DFA9A8
Type	
Requirement	Optional
Abstract:	
Summary:	

Determines the frequency with which temperature data is updated in milliseconds.

Examples	
Read	**Mandatory**
Write	**Mandatory**
Write Without Response	Excluded
Signed Write	Excluded
Reliable Write	Excluded
Notify	Excluded
Indicate	Excluded
Broadcast	Excluded
Writable Auxiliaries	Excluded
Extended Properties	Excluded
Descriptors	