BLUETOOTH® DOC	Date / Year-Month-Day 2012-08-21	Approved Adopted	Revision V10	Document No CSCS_SPEC
Prepared By SF WG	E-mail Address sf-main@bluetooth.org			N.B.

CYCLING SPEED AND CADENCE SERVICE

Abstract:

This service exposes speed-related and cadence-related data from a Cycling Speed and Cadence sensor intended for fitness applications.

Revision History

Revision	Date (yyyy-mm-dd)	Comments
D09r00	2012-03-28	Initial Draft, based on Running Speed and Cadence Service. Incorporated feedback from WG.
D09r01	2012-03-28	Accepted all changes.
D09r02	2012-04-03	Incorporated feedback from WG, accepted all changes. Incorporated ability to set cumulative wheel event value to any UINT32 value.
D09r03	2012-04-04	Accepted all changes. Prepared for BARB review. Added requirement in 3.1 and clarification in Table 3.3.
D09r04	2012-04-10	Accepted all changes. Submitted for BARB review. Incorporated feedback from Huanchun and Joe.
D09r05	2012-04-11	Accepted all changes. Resubmitted for BARB review. Incorporated further BARB feedback.
D09r06	2012-05-02	Accepted all changes. Submitted for BARB vote. Incorporated changes to align with updates to RSC service. Version used at IOP with changes attributed to “Author”.
D09r07	2012-05-02	Accepted all changes. Resubmitted for BARB review.
D10r00	2012-06-05	Changed revision to draft 1.0. Incorporated SF WG and BARB feedback.
D10r01	2012-06-28	Accepted all changes. Resubmitted for BARB review. Incorporated further feedback from BARB.
D10r02	2012-06-28	Accepted all changes. Resubmitted for final BARB review.
V10	2012-08-21	Adopted by the Bluetooth SIG Board of Directors

Contributors

Name	Company
Robert Hughes	Intel Corporation
Guillaume Schatz	Polar
Niclas Granqvist	Polar
Jari Miettinen	Polar
Chip Hawkins	Wahoo Fitness
Michael Moore	Wahoo Fitness

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*).

The term *Reserved for Future Use (RFU)* is used to indicate Bluetooth SIG assigned values that are reserved by the Bluetooth SIG and are not otherwise available for use by implementations.

Table of Contents

1 Introduction 6
1.1 Conformance 6
1.2 Service Dependency 6
1.3 Bluetooth Specification Release Compatibility 6
1.4 GATT Sub-Procedure Requirements 6
1.5 Transport Dependencies 6
1.6 Error Codes 7
1.7 Byte Transmission Order 7
2 Service Declaration 8
3 Service Characteristics 9
3.1 CSC Measurement 9
3.1.1 Characteristic Behavior 9
3.1.2 Characteristic Descriptors 12
3.2 CSC Feature 12
3.2.1 Characteristic Behavior 12
3.3 Sensor Location 13
3.3.1 Characteristic Behavior 13
3.4 SC Control Point 13
3.4.1 SC Control Point Procedure Requirements 14
3.4.2 SC Control Point Behavioral Description 14
3.4.3 General Error Handling procedures 15
3.4.4 Procedure Timeout 16
3.4.5 Characteristic Descriptors 16
3.5 Requirements for Time-Sensitive Data 16
4 SDP Interoperability 17
5 Acronyms and Abbreviations 18
6 References 19

1 Introduction

The Cycling Speed and Cadence (CSC) Service exposes speed-related data and/or cadence-related data while using the Cycling Speed and Cadence sensor (Server).

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.

1.4 GATT Sub-Procedure Requirements

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

Table 1.1 summarizes additional GATT sub-procedure requirements beyond those required by all GATT Servers.

GATT Sub-Procedure	Requirements
Write Characteristic Value	C.1
Notifications	M
Indications	C.1
Read Characteristic Descriptors	M
Write Characteristic Descriptors	M

Table 1.1: GATT Sub-procedure Requirements

C.1: Mandatory if the SC Control Point characteristic is supported, otherwise excluded for this service.

1.5 Transport Dependencies

There are no transport restrictions imposed by this service specification.

Where the term BR/EDR is used throughout this document, this also includes the use of AMP.

1.6 Error Codes

This service defines the following Attribute Protocol Application Error codes:

Name	Error Code	Description
Procedure Already in Progress	0x80	A SC Control Point request cannot be serviced because a previously triggered SC Control Point operation is still in progress.
Client Characteristic Configuration descriptor improperly configured	0x81	The Client Characteristic Configuration descriptor is not configured according to the requirements of the service.

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 [2].

2 Service Declaration

The Cycling Speed and Cadence Service is recommended to be instantiated as a «Primary Service».

The service UUID shall be set to «Cycling Speed and Cadence Service» defined in [2].

3 Service Characteristics

The following characteristics are exposed in the Cycling Speed and Cadence Service. Only one instance of each characteristic is permitted within this service.

Characteristic Name	Requirement	Mandatory Properties	Optional Properties	Security Permissions
CSC Measurement	M	Notify		None.
CSC Measurement Client Characteristic Configuration descriptor	M	Read, Write		None.
CSC Feature	M	Read		None.
Sensor Location	C.1	Read		None.
SC Control Point	C.2	Write, Indicate		None.
SC Control Point Client Characteristic Configuration Descriptor	C.2	Read, Write		None.

Table 3.1: Cycling Speed and Cadence Service characteristics

C.1: Mandatory if the Multiple Sensor Locations feature is supported, otherwise optional.

C.2: Mandatory if at least one SC Control Point procedure is supported, 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 CSC Measurement

The CSC Measurement characteristic is used to send speed-related data and/or cadence-related data. Included in the characteristic value are a Flags field (for showing the presence of optional fields) and depending upon the contents of the Flags field, either one or both of the following field pairs: Cumulative Wheel Revolutions and Last Wheel Event Time fields, and Cumulative Crank Revolutions and Last Crank Event Time fields.

The Server measures the speed-related data at which the bike is moving and cadence-related data which represents the number of times per minute the user turns the crank.

3.1.1 Characteristic Behavior

When the CSC Measurement characteristic is configured for notification via the *Client Characteristic Configuration* descriptor and a speed and cadence measurement is available, this characteristic shall be notified while in a connection.

Either the Wheel Revolution Data (containing the Cumulative Wheel Revolutions and Last Wheel Event Time fields) or the Crank Revolution Data (containing the Cumulative Crank Revolutions and Last Crank Event Time fields) or both shall be present in the CSC Measurement characteristic.

The CSC 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 CSC Measurement characteristic.

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

The bits of the Flags field are defined in the following subsections.

3.1.1.1.1 Wheel Revolution Data Present bit

The Wheel Revolution Data Present bit (bit 0 of the Flags field) indicates whether or not the Cumulative Wheel Revolutions and Last Wheel Event Time fields are present.

If the Wheel Revolution Data feature is not supported (see Section 3.2.1), then wheel revolution data (see Section 3.1.1.2) cannot be present and Wheel Revolution Data Present bit shall be 0.

If the Wheel Revolution Data feature is supported, then wheel revolution data may be present and the value of the Wheel Revolution Data Present bit shall be set to correspond to the presence of wheel revolution data in the CSC Measurement characteristic. In this case, the Wheel Revolution Data Present bit may change during a connection.

When the Cumulative Wheel Revolutions and Last Wheel Event Time fields are not present, the Wheel Revolution Data Present bit shall be set to 0. When the Cumulative Wheel Revolutions and Last Wheel Event Time fields are present, the Wheel Revolution Data Present bit shall be set to 1.

3.1.1.1.2 Crank Revolution Data Present bit

The Crank Revolution Data Present bit (bit 1 of the Flags field) indicates whether or not the Cumulative Crank Revolutions and the Last Crank Event Time fields are present.

If the Crank Revolution Data feature is not supported (see Section 3.2.1), then crank revolution data (see Section 3.1.1.3) cannot be present and Crank Revolution Data Present bit shall be 0.

If the Crank Revolution Data feature is supported, then crank revolution data may be present and the value of the Crank Revolution Data Present bit shall be set to correspond to the presence of crank revolution data in the CSC Measurement characteristic. In this case, the Crank Revolution Data Present bit may change during a connection.

When the Cumulative Crank Revolutions and the Last Crank Event Time fields are not present, the Crank Revolution Data Present bit shall be set to 0. When the Cumulative Crank Revolutions and the Last Crank Event Time fields are present, the Crank Revolution Data Present bit shall be set to 1.

3.1.1.2 Cumulative Wheel Revolutions and Last Wheel Event Time Fields

The Cumulative Wheel Revolutions value and the Last Wheel Event Time value allow the Client to calculate the speed (instantaneous and average) as well as the distance travelled. This calculation requires the Client to know the wheel circumference of the wheel where the measurement is taken.

If the Wheel Revolution Data Present bit is set to 1 (bit 0 of the Flags field), then the Cumulative Wheel Revolutions field and the Last Wheel Event Time field shall be present in the CSC Measurement characteristic. Otherwise, both fields shall not be present and bit 0 of the Flags field shall be set to 0.

The Cumulative Wheel Revolutions value, which represents the number of times a wheel rotates, is used in combination with the Last Wheel Event Time and the wheel circumference stored on the Client to determine 1) the speed of the bicycle and 2) the distance traveled. In addition, if there is link loss, the Cumulative Wheel Revolutions value can be used to calculate the average speed of the bicycle during the link loss. This value is expected to be set to 0 (or another desired value in case of e.g. a sensor upgrade) at initial installation on a bicycle as described in Section 3.4.2.1. The Cumulative Wheel Revolutions value may decrement for some implementations (e.g. If the bicycle is rolled in reverse), but shall not decrease below 0.

The ‘wheel event time’ is a free-running-count of 1/1024 second units and it represents the time when the wheel revolution was detected by the wheel rotation sensor. Since several wheel events can occur between transmissions, only the Last Wheel Event Time value is transmitted. This value is used in combination with the Cumulative Wheel Revolutions value to enable the Client to calculate speed and distance.

Since Cumulative Wheel Revolutions value is a UINT32, the highest value that can be represented is 4,294,967,296 revolutions. Assuming a wheel circumference of 2.1 meters, the maximum distance that can be represented is 9,019,431 kilometers. Since the product life expectancy of CSC Sensor is about 5 years and given that top level cyclists may reach 15,000 kilometer a year (75,000 km in 5 years), this value significantly exceeds the expectation. This value is not permitted to roll over. If a reset or other specific setting of the Cumulative Wheel Revolutions value is required, see Section 3.4.2.1 for requirements related to setting the value of this field.

The Last Wheel Event Time value rolls over every 64 seconds.

3.1.1.3 Cumulative Crank Revolutions and Last Crank Event Time Fields

The Cumulative Crank Revolutions value and the Last Crank Event Time value allow the Client to calculate the cadence (instantaneous and average) at which the user turns the crank.

If the Crank Revolution Data Present bit is set to 1 (bit 1 of the Flags field), then the Cumulative Crank Revolutions field and the Last Crank Event Time field shall be present in the CSC Measurement characteristic. Otherwise, both fields shall not be present and bit 1 of the Flags field shall be set to 0.

The Cumulative Crank Revolutions value, which represents the number of times a crank rotates, is used in combination with the Last Crank Event Time to determine 1) if the cyclist is coasting and 2) the average cadence. Average cadence is not accurate unless 0 cadence events (i.e. coasting) are subtracted. In addition, if there is link loss, the Cumulative Crank Revolutions value can be used to calculate the average cadence during the link loss. This value is intended to roll over and is not configurable.

The ‘crank event time’ is a free-running-count of 1/1024 second units and it represents the time when the crank revolution was detected by the crank rotation sensor. Since several crank events can occur between transmissions, only the Last Crank Event Time value is transmitted. This value is used in combination with the Cumulative Crank Revolutions value to enable the Client to calculate cadence.

The Last Crank Event Time value rolls over every 64 seconds.

To enhance the user experience, the Server should not take into account the extra crank revolutions that may be detected when the user is not pedaling (e.g. coasting down the hill) but the sensor is facing the magnet installed on the crankset and may cause unwanted crank revolution detections.

3.1.1.4 Transmission Interval

In typical applications, the CSC Measurement characteristic is notified approximately once per second. This interval may vary and is determined by the Server and not required to be configurable by the Client.

3.1.2 Characteristic Descriptors

3.1.2.1 Client Characteristic Configuration Descriptor

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

3.2 CSC Feature

The CSC Feature characteristic shall be used to describe the supported features of the Server.

Reserved for Future Use (RFU) bits in the CSC Feature characteristic value shall be set to 0.

3.2.1 Characteristic Behavior

When read, the CSC Feature characteristic returns a value that is used by a Client to determine the supported features of the Server.

The bits of the CSC Feature characteristic may either be static for the lifetime of the device (i.e. static permanently or until Service Changed is indicated) or guaranteed to be static only during a connection. This requirement is defined in the table below on a bit-by-bit basis. Although all defined bits in this version of this specification are required to be static during the lifetime of a device, it is possible that some future bits will be defined as being static only during a connection.

Bit	CSC Feature Bit	Static Requirement
0	Wheel Revolution Data Supported	Lifetime
1	Crank Revolution Data Supported	Lifetime
2	Multiple Sensor Locations Supported	Lifetime
3-15	Reserved for Future Use	Not defined.

Table 3.2: Static Requirements for CSC Feature Bits

If the Wheel Revolution Data feature is not supported, the Wheel Revolution Data Supported bit shall be set to 0 and the Wheel Revolution Data Present bit (from the Flags field of the CSC Measurement characteristic) shall also be set to 0. Otherwise the Wheel Revolution Data Supported bit shall be set to 1 and the Wheel Revolution Data Present bit shall be used to show whether or not Cumulative Wheel Revolutions and Last Wheel Event Time fields are present in the CSC Measurement characteristic. See Section 3.1.1.2 for more information in these fields.

If the Crank Revolution Data feature is not supported, the Crank Revolution Data Supported bit shall be set to 0 and the Crank Revolution Data Present bit (from the Flags field of the CSC Measurement characteristic) shall also be set to 0. Otherwise the Crank Revolution Data Supported bit shall be set to 1 and the Crank Revolution Data Present bit shall be used to show whether or not Cumulative Crank Revolutions and Last Crank Event Time fields are present in the CSC Measurement characteristic. See Section 3.1.1.3 for more information in these fields.

If the Multiple Sensor Locations feature is not supported, the Multiple Sensor Locations Supported bit shall be set to 0. Otherwise the Multiple Sensor Locations Supported bit shall be set to 1 (Multiple Sensor Locations feature supported).

3.3 Sensor Location

The Sensor Location characteristic of the device may be used to describe the physical location of the Server when correctly fitted.

If the Server supports the Multiple Sensor Locations feature, the value of the Sensor Location characteristic may be updated while in a connection as described in Section 3.4.2.2. Otherwise, if the Sensor Location characteristic is present and the Multiple Sensor Locations feature is not supported, the value of the Sensor Location shall be static for the lifetime of the Server or until Service Changed characteristic is indicated.

If the Server supports the Multiple Sensor Locations feature, the Client should not assume that the value of the Sensor Location characteristic of the Server is set to the same value as at the end of a previous connection. This is primarily because the value may have been altered by a different Client after the previous connection (e.g. the user has moved his sensor to another location and configured the new Sensor Location with another Client).

3.3.1 Characteristic Behavior

The Sensor Location characteristic returns the sensor location value when read.

3.4 SC Control Point

If the SC Control Point is supported, profiles utilizing this service are required to ensure that the Client configures the SC Control Point characteristic for indications (i.e. via the *Client Characteristic Configuration* descriptor) at the first connection.

Support for this characteristic is mandatory if the Server supports Wheel Revolution Data or Multiple Sensor Locations features, otherwise it is excluded for this version of the service in accordance with Table 3.1.

3.4.1 SC Control Point Procedure Requirements

Table 3.3 shows the requirements for the SC Control Point characteristic in the context of this service:

Procedure	Requirement	Properties	Parameter Description	Applicable Response Value	Response Parameter
Set Cumulative Value	C.1	Write	Cumulative Value (UINT32)	Success, Operation Failed Op Code Not Supported	None
Start Sensor Calibration	Not used in this version of the specification.	N/A	N/A	N/A	N/A
Update Sensor Location	C.2	Write	Sensor Location (UINT8)	Success, Operation Failed, Invalid Parameter Op Code Not Supported	None
Request Supported Sensor Locations	C.2	Write	None	Success	Byte array – see 3.4.2.3
				Operation Failed Op Code Not Supported	None

Table 3.3: SC Control Point Procedure Requirements

C.1: Mandatory if Wheel Revolutions Data feature is supported, otherwise excluded from this version of this Service.

C.2: Mandatory if Multiple Sensor Locations feature is supported, otherwise excluded from this version of this Service.

3.4.2 SC Control Point Behavioral Description

The SC Control Point is used by a Client to control certain behaviors of the Server. Procedures are triggered by a Write to this characteristic value that includes an Op Code specifying the operation (see Table 3.3) which may be followed by a Parameter that is valid within the context of that Op Code.

3.4.2.1 Set Cumulative Value Procedure

When the *Set Cumulative Value* Op Code is written to the SC Control Point and if the Wheel Revolution Data feature is supported by the Server, the Server shall set the Cumulative Wheel Revolutions value to the same value as the UINT32 parameter which accompanies the op code. For example, a parameter of 0x00000000 will set the Cumulative Wheel Revolutions value to 0. The response shall be indicated when the Cumulative Wheel Revolutions value is applied using the *Response Code* Op Code, the *Request Op Code* along with “Success” or other appropriate *Response Value*.

If the operation results in an error condition or if the Wheel Revolution Data feature is not supported by the Server, see Section 3.4.3 for details on handling this condition.

This procedure shall not be used to set the Cumulative Crank Revolutions value since this value is not configurable.

3.4.2.2 Update Sensor Location Procedure

When the *Update Sensor Location* Op Code is written to the SC Control Point and if the Multiple Sensor Locations feature is supported by the Server, the Server shall update the value of the Sensor Location characteristic with the value of the desired sensor location transmitted as a Parameter of the SC Control Point. The response shall be indicated when the sensor location is updated in the Server using the *Response Code* Op Code, the *Request Op Code* along with “Success” or other appropriate *Response Value*.

The Server should cache the most recent value of the Sensor Location characteristic to avoid reconfiguration of this characteristic by the Client each time a connection is established.

If the operation results in an error condition, or if the Multiple Sensor Locations feature is not supported by the Server, see Section 3.4.3 for details on handling this condition.

3.4.2.3 Request Supported Sensor Locations Procedure

When the *Request Supported Sensor Locations* Op Code is written to the SC Control Point and if the Multiple Sensor Locations feature is supported by the Server, the Server shall send a list of the supported sensor location values (i.e. a byte array containing values of each supported sensor location). The response shall be indicated using the *Response Code* Op Code, the *Request Op Code*, the appropriate *Response Value* and, if the procedure succeeds, the *Response Value* shall be set to “Success” followed by a list of supported sensor location values in the Response Parameter as described in [2].

For a default ATT MTU, a maximum of 17 supported sensor locations can be sent.

If the operation results in an error condition or if the Multiple Sensor Locations feature is not supported by the Server, see Section 3.4.3 for details on handling this condition.

3.4.3 General Error Handling procedures

Other than error handling procedures that are specific to certain Op Codes, the following apply:

If an Op Code is written to the SC Control Point characteristic that is unsupported by the Server, the Server, after sending a Write Response, shall indicate the SC Control Point with a *Response Code* Op Code, the *Request Op Code* and *Response Value* set to *Op Code Not Supported*.

If a Parameter is written to the SC Control Point characteristic that is invalid (e.g. the Client writes the *Update Sensor Location* Op Code with a sensor location that is not valid in the context of the Server), the Server, after sending a Write Response, shall indicate the SC Control Point with a *Response Code* Op Code, the *Request Op Code* and *Response Value* set to *Invalid Parameter*.

If an Op Code is written to the SC Control Point characteristic while the Server is performing a previously triggered SC Control Point operation (i.e. resulting from invalid Client behavior), the Server shall return an error response with the Attribute Protocol Application error code set to *Procedure Already In Progress* as defined in Section 1.6.

If an Op Code is written to the SC Control Point characteristic and the Client Characteristic Configuration descriptor of the SC Control Point is not configured for indications, the Server shall return an error response with the Attribute Protocol Application error code set to *Client Characteristic Configuration Descriptor Improperly Configured* as defined in Section 1.6.

3.4.4 Procedure Timeout

In the context of the SC Control Point characteristic, a procedure is started when a write to the SC Control Point characteristic is successfully completed. When a procedure is complete, the Server shall indicate the SC Control Point with the Op Code set to *Response Code*.

In the context of the SC Control Point characteristic, a procedure is not considered started and not queued in the Server when a write to the SC Control Point results in an error response with the Attribute Protocol error code defined in Section 1.6.

When the Server transmits an indication of a characteristic, the acknowledgement response shall be considered to have timed out if a handle/value confirmation is not received from the Client within the ATT transaction timeout, defined as 30 seconds in Volume 2 Part F Section 3.3.3 of [1]. If a timeout occurs, the Server shall stop sending any further indications and notifications related to the operation and consider the procedure to have failed and may disconnect.

3.4.5 Characteristic Descriptors

3.4.5.1 Client Characteristic Configuration Descriptor

The *Client Characteristic Configuration* descriptor shall be included with the SC Control Point characteristic.

3.5 Requirements for Time-Sensitive Data

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

Since this service provides only a time stamp without a reference to identify the measurement time (and age) of the data and not a time stamp with a reference, the value of the CSC Measurement characteristic shall be discarded if either the connection does not get established or if the notification is not successfully transmitted (e.g., due to link loss).

4 SDP Interoperability

If this service is exposed over BR/EDR then it shall have the following SDP record.

Item	Definition	Type	Value	Status
Service Class ID List				M
Service Class #0		UUID	«Cycling Speed and Cadence»	M
Protocol Descriptor List				M
Protocol #0		UUID	L2CAP	M
Parameter #0 for Protocol #0	PSM	Uint16	PSM = ATT	M
Protocol #1		UUID	ATT	M
Parameter #0 for Protocol #1	GATT Start Handle	Uint16	First handle of this service in the GATT database	M
Parameter #1 for Protocol #1	GATT End Handle	Uint16	Last handle of this service in the GATT database	M
BrowseGroupList			PublicBrowseRoot*	M

Table 4.1: SDP Record

* PublicBrowseRoot shall be present; however, other browse UUIDs may also be included in the list.

5 Acronyms and Abbreviations

Acronyms and Abbreviations	Meaning
AMP	Alternate MAC PHY
BR/EDR	Basic Rate / Enhanced Data Rate
CSC	Cycling Speed and Cadence
GAP	Generic Access Profile
GATT	Generic Attribute Profile
LE	Low Energy
RFU	Reserved for Future Use
SC	Speed and Cadence
SDP	Service Discovery Protocol
UUID	Universally Unique Identifier

Table 5.1: Acronyms and Abbreviations

6 References

[1] Bluetooth Core Specification v4.0
[2] Characteristic and Descriptor descriptions are accessible via the Bluetooth SIG Assigned Numbers.