Next DST Change Service (NDCS)

Application Programming Interface Reference Manual

Profile Version: 1.0

Release: 4.0.1
January 10, 2014

Louisville, KY www.stonestreetone.com

Table of Contents

1. Introduction

Bluetopia®+LE is Stonestreet One's Bluetooth protocol stack that supports the adopted Bluetooth low energy specification. Stonestreet One's upper level protocol stack that supports Single Mode devices is Bluetopia®+LE Single. More specifically, this stack is a software solution that resides above the Physical HCI (Host Controller Interface) Transport Layer and extends through the L2CAP (Logical Link Control and Adaptation Protocol), ATT (Attribute Protocol) Link Layers, the GAP (Generic Attribute Profile) Layer and the Genetic Attribute Protocol (GATT) Layer. In addition to basic functionality of these layers, the Bluetooth Protocol Stack by Stonestreet One provides implementations of the Device Information Service (DIS), NDCS (Next DST Change Service), and several of the Bluetooth Profiles. Program access to these layers, services, and profiles is handled via Application Programming Interface (API) calls.

The remainder of this chapter has sections on the scope of this document, other documents applicable to this document, and a listing of acronyms and abbreviations. Chapter 2 is the API reference that contains a description of all programming interfaces for the Next DST Change Service Profile Stack provided by Bluetopia®+LE Single. And, Chapter 3 contains the header file name list for the Next DST Change Servicer Profile library.

1.1 Scope

This reference manual provides information on the APIs identified in Figure 1-1 below. These APIs are available on the full range of platforms supported by Stonestreet One:

- Windows
- Windows Mobile
- Windows CE
- Linux
- QNX
- Other Embedded OS

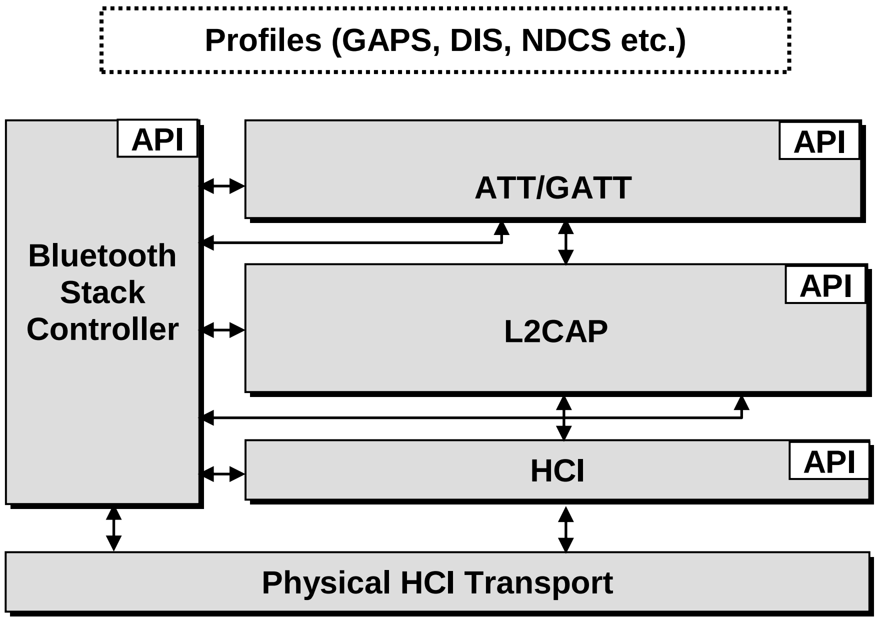

Figure 1-1 The Stonestreet One Bluetooth Protocol Stack

1.2 Applicable Documents

The following documents may be used for additional background and technical depth regarding the Bluetooth technology.

1. *Specification of the Bluetooth System, Volume 1, Architecture and Terminology Overview*, version 4.0, June 30, 2010.

2. *Specification of the Bluetooth System, Volume 6, Core System Package [Low Energy Controller Volume]*, version 4.0, June 30, 2010.

3. *Bluetopia® Protocol Stack, Application Programming Interface Reference Manual,* version 4.0.1, January 10, 2013.

4. *Bluetooth Doc Next DST Change Service Specification*, version v10r00, September 15, 2011

Possible error returns are listed for each API function call. These are the *most likely* errors, but in fact programmers should allow for the possibility of any error listed in the BTErrors.h header file to occur as the value of a function return.

1.3 Acronyms and Abbreviations

Acronyms and abbreviations used in this document and other Bluetooth specifications are listed in the table below.

Term	Meaning
API	Application Programming Interface
ATT	Attribute Protocol
NDCS	Next DST Change Service
BD_ADDR	Bluetooth Device Address
BT	Bluetooth
DIS	Device Information Service
GATT	Generic Attribute Protocol
GAPS	Generic Access Profile Service
HCI	Host Controller Interface
HS	High Speed
L2CAP	Logical Link Control and Adaptation Protocol
LE	Low Energy

2. Next DST Change Service Programming Interfaces

The Next DST Change Service programming interface defines the protocols and procedures to be used to implement Next DST Change Service capabilities. The Next DST Change Service commands are listed in section 2.1, the event callback prototypes are described in section 2.2, and the Next DST Change Service events are itemized in section 2.3. The actual prototypes and constants outlined in this section can be found in the **NDCSAPI.H** header file in the Bluetopia distribution.

2.1 Next DST Change Service Commands

The available Next DST Change Service command functions are listed in the table below and are described in the text that follows.

Function	Description
NDCS_Initialize_Service	Opens a NDCS Server.
NDCS_Cleanup_Service	Closes an opened NDCS Server.
NDCS_Initialize_Service_Handle_Range	Opens a NDCS Server with the ability to control the location of the service in the GATT database.
NDCS_Time_With_DST_Read_Request_Response	Responds to a read Next DST Change Time request from the remote device.
NDCS_Query_Number_Attributes	Queries the number of attributes.
NDCS_Time_With_DST_Read_Request_Error_Response	On Error, Responds to a read Next DST Change Time request from the remote device.
NDCS_Decode_Time_With_DST	Parses the value received from a remote NDCS Server interpreting it as Time with DST characteristic of NDCS Service.

NDCS_Initialize_Service

This function opens a NDCS Server on a specified Bluetooth Stack.

Prototype:

int BTPSAPI **NDCS_Initialize_Service** (unsigned int BluetoothStackID, NDCS_Event_Callback_t EventCallback, unsigned long CallbackParameter, unsigned int *ServiceID);

Parameters:

BluetoothStackID	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
EventCallback	Callback function that is registered to receive events that are associated with the specified service.

CallbackParameter	A user-defined parameter that will be passed back to the user in the callback function.
ServiceID	Unique GATT Service ID of the registered NDCS service returned from GATT_Register_Service API

Return:

Positive, non-zero if successful. The return value will be the Service Instance ID of NDCS Server that was successfully opened on the specified Bluetooth Stack ID. *This* is the value that should be used in all subsequent function calls that require Instance ID.

An error code if negative; one of the following values:

NDCS_ERROR_INSUFFICIENT_RESOURCES
NDCS_ERROR_SERVICE_ALREADY_REGISTERED
NDCS_ERROR_INVALID_PARAMETER
BTGATT_ERROR_INVALID_SERVICE_TABLE_FORMAT
BTGATT_ERROR_INSUFFICIENT_RESOURCES
BTGATT_ERROR_INVALID_PARAMETER
BTGATT_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGATT_ERROR_NOT_INITIALIZED

Possible Events:

NDCS_Initialize_Service_Handle_Range

This function is responsible for opening a NDCS Server with the ability to control the location of the service in the GATT database.

Prototype:

int BTPSAPI **NDCS_Initialize_Service_Handle_Range**(unsigned int BluetoothStackID,
NDCS_Event_Callback_t EventCallback, unsigned long CallbackParameter,
unsigned int *ServiceID, GATT_Attribute_Handle_Group_t *ServiceHandleRange);

Parameters:

BluetoothStackID	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
EventCallback	Callback function that is registered to receive events that are associated with the specified service.
CallbackParameter	A user-defined parameter that will be passed back to the user in the callback function.
ServiceID	Unique GATT Service ID of the registered NDCS service returned from GATT_Register_Service API
ServiceHandleRange	Pointer to a Service Handle Range structure, that on input can be used to control the location of the service in the GATT database, and on output returns the handle range that the service is using in the GATT database.

Return:

Positive, non-zero if successful. The return value will be the Service Instance ID of NDCS Server that was successfully opened on the specified Bluetooth Stack ID. This is the value that should be used in all subsequent function calls that require Instance ID.

An error code if negative; one of the following values:

NDCS _ERROR_INSUFFICIENT_RESOURCES
NDCS _ERROR_INVALID_PARAMETER
GAPS_ERROR_SERVICE_ALREADY_REGISTERED
BTGATT_ERROR_INVALID_SERVICE_TABLE_FORMAT
BTGATT_ERROR_INSUFFICIENT_RESOURCES
BTGATT_ERROR_INVALID_PARAMETER
BTGATT_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGATT_ERROR_NOT_INITIALIZED

Possible Events:

NDCS_Cleanup_Service

This function is responsible for cleaning up and freeing all resources associated with a NDCS Service Instance. After this function is called, no other NDCS Service function can be called until after a successful call to the NDCS_Initialize_Service() function is performed.

Prototype:

int BTPSAPI **NDCS_Cleanup_Service**(unsigned int BluetoothStackID, unsigned int InstanceID);

Parameters:

BluetoothStackID	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
InstanceID	The Service Instance ID to close. This is the value that was returned from the NDCS_Initialize_Service() function.

Return: Zero if successful. An error code if negative; one of the following values:

NDCS_ERROR_INVALID_PARAMETER
NDCS_ERROR_INVALID_INSTANCE_ID

Possible Events:

NDCS_Query_Number_Attributes

This function is responsible for querying the number of attributes that are contained in the NDCS Service that is registered with a call to NDCS _Initialize_Service().

Prototype:

unsigned int BTPSAPI **NDCS_Query_Number_Attributes**(void)

Parameters:

Return: Zero if successful.

An error code if negative; one of the following values:

NDCS _ERROR_INVALID_PARAMETER
NDCS _ERROR_INVALID_INSTANCE_ID

Possible Events:

NDCS_Time_With_DST _Read_Request_Response

This function is responsible for responding to Next DST Change Time read request to remote device.

Prototype:

int BTPSAPI **NDCS_Time_With_DST_Read_Request_Response**(unsigned int BluetoothStackID, unsigned int TransactionID, NDCS_Time_With_Dst_Data_t *Next_Dst_Change_Time);

Parameters:

BluetoothStackID	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
TransactionID	Transaction ID of the original read request. This value was received in the etNDCS_Server_Read_Time_With_Dst_Request event.
Next_Dst_Change_Time	Specifies the Next DST Change Time to send to remote device.

Return:

Zero if successful.

An error code if negative; one of the following values:

NDCS_ERROR_INVALID_PARAMETER
BTGATT_ERROR_INVALID_TRANSACTION_ID
BTGATT_ERROR_NOT_INITIALIZED
BTGATT_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGATT_ERROR_INVALID_PARAMETER

Possible Events:

etNDCS_Server_Time_With_Dst_Request

NDCS_Time_With_DST_Read_Request_Error_Response

This function is responsible for responding to Next DST Change Time read request when an error occurred.

Prototype:

int BTPSAPI **NDCS_Time_With_DST_Read_Request _Error_Response**(unsigned int BluetoothStackID, unsigned int TransactionID, Byte_t ErrorCode);

Parameters:

BluetoothStackID	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
TransactionID	Transaction ID of the original read request. This value was received in the etNDCS_Server_Read_Time_With_Dst_Request event.
ErrorCode	ErrorCode occurred during read operation

Return:

Zero if successful.

An error code if negative; one of the following values:

NDCS_ERROR_INVALID_PARAMETER
BTGATT_ERROR_INVALID_TRANSACTION_ID
BTGATT_ERROR_NOT_INITIALIZED
BTGATT_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGATT_ERROR_INVALID_PARAMETER

Possible Events:

etNDCS_Server_Read_Time_With_Dst_Request

NDCS_Decode_Time_With_DST

This function is responsible for parsing a value received from a remote NDCS Server interpreting it as Time with DST characteristic value.

Prototype:

int BTPSAPI **NDCS_Decode_Time_With_DST**(unsigned int ValueLength, Byte_t *Value, NDCS_Time_With_Dst_Data_t *Next_Dst_Change_Time);

Parameters:

ValueLength	Specifies the length of the value returned by the remote NDCS Server.
Value	Value is a pointer to the data returned by the remote NDCS Server.
Next_Dst_Change_Time	A pointer to store the parsed Time with DST characteristic value.

Return:

Zero if successful.

An error code if negative; one of the following values:

NDCS_ERROR_MALFORMATTED_DATA

Possible Events:

2.2 Next DST Change Service Event Callback Prototypes

2.2.1 Server Event Callback

The event callback function mentioned in the NDCS_Initialize_Service command accepts the callback function described by the following prototype.

NDCS_Event_Callback_t

Prototype of callback function passed in the NDCS_Initialize_Service command.

Prototype:

typedef void (BTPSAPI ***NDCS_Event_Callback_t**)(unsigned int BluetoothStackID, NDCS_Event_Data_t *NDCS_Event_Data, unsigned long CallbackParameter);

Parameters:

BluetoothStackID	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
NDCS_Event_Data_t	Data describing the event for which the callback function is called. This is defined by the following structure:

```
typedef struct _tagNDCS_Event_Data_t
{
  NDCS_Event_Type_t Event_Data_Type;
  Word_t            Event_Data_Size;
  union
  {
   NDCS_Read_Time_With_DST_Request_Data_t
                           *NDCS_Read_Time_With_DST_Request_Data;
  } Event_Data;
} NDCS_Event_Data_t;
```

Where, Event_Data_Type is one of the enumerations of the event types listed in the table in section 2.3, and each data structure in the union is described with its event in that section as well.

CallbackParameter	User-defined parameter that was defined in the callback registration.

Return:

2.3 Next DST Change Service Events

The Next DST Change Service contains events that are received by the Server. The following sections detail those events.

2.3.1 Next DST Change Service Server Events

The possible Next DST Change Service Server Events from the Bluetooth stack are listed in the table below and are described in the text which follows:

Event	Description
etNDCS_Server_Read_Time_With_Dst_Request	Dispatched when a NDCS Client requests read Time with DST request to a registered NDCS Server.

etNDCS_Server_Read_Time_With_Dst_Request

Dispatched when a NDCS Client requests Time with DST read request to a registered NDCS Server.

Return Structure:

```
typedef struct _tagNDCS_Read_Time_With_DST_Request_Data_t
{
   unsigned int              InstanceID;
   unsigned int              ConnectionID;
   unsigned int              TransactionID;
   GATT_Connection_Type_t    ConnectionType;
   BD_ADDR_t                 RemoteDevice;
} NDCS_Read_Time_With_DST_Request_Data_t;
```

Event Parameters:

InstanceID	Identifies the Local Server Instance to which the Remote Client has connected.
ConnectionID	Identifier that uniquely identifies the actual connection of remote device that is making the request.
Transaction ID	Specifies the unique Transaction ID of remote device that is making the request.
ConnectionType	Identifies the type of remote Bluetooth device that is connected. Currently this value will be gctLE only.
RemoteDevice	Specifies the address of the Client Bluetooth device that has connected to the specified Server.

3. File Distributions

The header files that are distributed with the Bluetooth Next DST Change Service Library are listed in the table below.

File	Contents/Description
NDCSAPI.h	Bluetooth Next DST Change Service (GATT based) API Type Definitions, Constants, and Prototypes.
NDCSType.h	Bluetooth Next DST Change Service Types.
SS1BTNDC.h	Bluetooth Next DST Change Service Include file