Drake Control Service

Base UUID: 4F770000ED7D-11E4-840E-0002A5D5C51B

Service UUID: 0x0601

Abstract:

This service exposes board specific control methods .

Summary:

TODO

Service Dependencies:

This service is not dependent upon any other service.

GATT Requirements

Sub-Procedure	Server Requirement
Write Characteristic Value	Mandatory
Notifications	Excluded
Indications	Mandatory
Read Characteristic Descriptors	Mandatory
Write Characteristic Descriptors	Mandatory

Transport Dependencies

Transport	Supported
Classic	false
Low Energy	true
High Speed	false

Error Codes

Name	Code	Description

Service Characteristics

<table>
<tr><th>Overview</th><th>Properties</th><th>Security</th><th>Descriptors</th></tr>
<tr>
<td>Name:
Drake Status
Requirement:
Optional</td>
<td>
<table>
<tr><th>Property</th><th>Requirement</th></tr>
<tr><td>Read</td><td>Mandatory</td></tr>
<tr><td>Write</td><td>Excluded</td></tr>
<tr><td>WriteWithoutResponse</td><td>Excluded</td></tr>
<tr><td>SignedWrite</td><td>Excluded</td></tr>
<tr><td>Notify</td><td>Mandatory</td></tr>
<tr><td>Indicate</td><td>Excluded</td></tr>
<tr><td>WriteableAuxiliaries</td><td>Excluded</td></tr>
<tr><td>Broadcast</td><td>Excluded</td></tr>
<tr><td>ExtendedProperties</td><td></td></tr>
</table>
</td>
<td>None</td>
<td>
<table>
<tr><th>Overview</th><th>Permissions</th></tr>
<tr><td>Name:
Client Characteristic Configuration
Requirement:
Mandatory</td><td>
<table>
<tr><th>Perm.</th><th>Req.</th></tr>
<tr><td>Read</td><td>Mandatory</td></tr>
<tr><td>Write</td><td>Mandatory</td></tr>
</table>
</td></tr>
</table>
</td>
</tr>
<tr>
<td>Name:
Drake Feature
Requirement:
Mandatory</td>
<td>
<table>
<tr><th>Property</th><th>Requirement</th></tr>
<tr><td>Read</td><td>Mandatory</td></tr>
<tr><td>Write</td><td>Excluded</td></tr>
<tr><td>WriteWithoutResponse</td><td>Excluded</td></tr>
<tr><td>SignedWrite</td><td>Excluded</td></tr>
<tr><td>Notify</td><td>Excluded</td></tr>
<tr><td>Indicate</td><td>Excluded</td></tr>
<tr><td>WriteableAuxiliaries</td><td>Excluded</td></tr>
<tr><td>Broadcast</td><td>Excluded</td></tr>
<tr><td>ExtendedProperties</td><td></td></tr>
</table>
</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Name:
Drake Control Point
Requirement:
Mandatory</td>
<td>
<table>
<tr><th>Property</th><th>Requirement</th></tr>
<tr><td>Read</td><td>Excluded</td></tr>
<tr><td>Write</td><td>Mandatory</td></tr>
<tr><td>WriteWithoutResponse</td><td>Excluded</td></tr>
<tr><td>SignedWrite</td><td>Excluded</td></tr>
<tr><td>Notify</td><td>Excluded</td></tr>
<tr><td>Indicate</td><td>Mandatory</td></tr>
<tr><td>WriteableAuxiliaries</td><td>Excluded</td></tr>
<tr><td>Broadcast</td><td>Excluded</td></tr>
<tr><td>ExtendedProperties</td><td></td></tr>
</table>
</td>
<td>Optional</td>
<td>
<table>
<tr><th>Overview</th><th>Permissions</th></tr>
<tr><td>Name:
Client Characteristic Configuration
Requirement:
Mandatory</td><td>
<table>
<tr><th>Perm.</th><th>Req.</th></tr>
<tr><td>Read</td><td>Mandatory</td></tr>
<tr><td>Write</td><td>Mandatory</td></tr>
</table>
</td></tr>
</table>
</td>
</tr>
</table>

Drake Status

Characteristic UUID: 0x0602

Summary:

The Drake Staus characteristic is a variable length structure containing a Flags field and, based on the contents of the Flags field, may contain one or more additional fields as shown in the table below.

Value Fields

<table>
<tr><th>Names</th><th>Field Req.</th><th>Format</th><th>Additional Information</th></tr>
<tr><td>Flags</td><td>Mandatory</td><td>8bit</td><td>Bit Field
<table>
<tr><th>Bit</th><th>Size</th><th>Name</th><th>Definition</th></tr>
<tr><td>0</td><td>1</td><td>plunger status present</td><td><table><tr><td>0</td><td>False</td></tr><tr><td>1</td><td>True</td></tr></table></td></tr>
<tr><td>1</td><td>7</td><td>reserved for future use</td><td></td></tr>
</table></td></tr>
<tr><td>Plunger Status
Information:
Unit is in percent with a resolution of 0.5
Unit:
org.bluetooth.unit.percentage
Exponent: Binary, -1</td><td>Optional</td><td>uint8</td><td>0% closed
100% fully opened</td></tr>
</table>

Drake Feature

Characteristic UUID: 0x0603

Summary:

The Drake Feature characteristic is used to report a list of features supported by the device.

Value Fields

<table>
<tr><th>Names</th><th>Field Req.</th><th>Format</th><th>Additional Information</th></tr>
<tr><td>Device Features</td><td>Mandatory</td><td>8bit</td><td>Bit Field
<table>
<tr><th>Bit</th><th>Size</th><th>Name</th><th>Definition</th></tr>
<tr><td>0</td><td>1</td><td>light supported</td><td><table><tr><td>0</td><td>False</td></tr><tr><td>1</td><td>True</td></tr></table></td></tr>
<tr><td>1</td><td>1</td><td>seat-post actuator supported</td><td><table><tr><td>0</td><td>False</td></tr><tr><td>1</td><td>True</td></tr></table></td></tr>
<tr><td>2</td><td>6</td><td>reserved for future use</td><td></td></tr>
</table></td></tr>
<tr><td>Light Features</td><td>Mandatory</td><td>8bit</td><td>Bit Field
<table>
<tr><th>Bit</th><th>Size</th><th>Name</th><th>Definition</th></tr>
<tr><td>0</td><td>1</td><td>front-light pattern supported</td><td><table><tr><td>0</td><td>False</td></tr><tr><td>1</td><td>True</td></tr></table></td></tr>
<tr><td>1</td><td>1</td><td>tail-light pattern supported</td><td><table><tr><td>0</td><td>False</td></tr><tr><td>1</td><td>True</td></tr></table></td></tr>
<tr><td>2</td><td>1</td><td>position-light pattern supported</td><td><table><tr><td>0</td><td>False</td></tr><tr><td>1</td><td>True</td></tr></table></td></tr>
<tr><td>3</td><td>1</td><td>brake-light pattern supported</td><td><table><tr><td>0</td><td>False</td></tr><tr><td>1</td><td>True</td></tr></table></td></tr>
<tr><td>4</td><td>1</td><td>left indicator-light pattern supported</td><td><table><tr><td>0</td><td>False</td></tr><tr><td>1</td><td>True</td></tr></table></td></tr>
<tr><td>5</td><td>1</td><td>right indicator-light pattern supported</td><td><table><tr><td>0</td><td>False</td></tr><tr><td>1</td><td>True</td></tr></table></td></tr>
<tr><td>6</td><td>2</td><td>reserved for future use</td><td></td></tr>
</table></td></tr>
<tr><td>Actuator Features</td><td>Mandatory</td><td>8bit</td><td>Bit Field
<table>
<tr><th>Bit</th><th>Size</th><th>Name</th><th>Definition</th></tr>
<tr><td>0</td><td>1</td><td>motor configuration supported</td><td><table><tr><td>0</td><td>False</td></tr><tr><td>1</td><td>True</td></tr></table></td></tr>
</table></td></tr>
</table>

			1	1	motor sensor supported	0 False 1 True
			2	1	travel sensor supported	0 False 1 True
			3	1	travel indexing supported	0 False 1 True
			4	4	reserved for future use	

Drake Control Point

Characteristic UUID: 0x0604

Summary:

The Drake Control Point characteristic is used to request a specific function to be executed on the receiving device.

Value Fields

<table>
<tr><th>Names</th><th>Field Req.</th><th>Format</th><th>Additional Information</th></tr>
<tr><td>Op codes</td><td>Mandatory</td><td>uint8</td><td>Enumerations
<table>
<tr><th>Key</th><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>request device usage</td><td>Request the devices main usage. The response to this control point is 0x20 followed by the devices usage.
See Table 1: Device usage enumeration for usage definitions.</td></tr>
<tr><td>2</td><td>set device usage</td><td>Initiate the procedure to set the devices main usage. The new usage is sent with preceding Op Code 0x02. The response to this control point is 0x20.
See Table 1: Device usage enumeration for usage definitions.</td></tr>
<tr><td>3</td><td>request battery information</td><td>Request the battery information. The response to this control point is 0x20 followed by the battery information See Table 2: Battery information for battery information definitions</td></tr>
<tr><td>4</td><td>set battery information</td><td>Initiate the procedure to set the battery information. The new information is sent with preceding Op Code 0x04. The response to this control point is 0x20. See Table 2: Battery information for battery information definitions</td></tr>
<tr><td>5</td><td>request soc at startup timeout</td><td>Request the state of charge at startup timeout. The response to this control point is 0x20 followed by the timeout in 1/1024s. This represents the length of one morse dot when device is used at dropper actuator and when device is used at light it represents the time in 1/1024s the indication is shown.</td></tr>
</table></td></tr>
</table>

			6	set soc at startup timeout	Initiate the procedure to set the state of charge at startup timeout. The new timeout is sent with preceding Op Code 0x06. The response to this control point is 0x20. This represents the length of one morse dot when device is used at dropper actuator and when device is used at light it represents the time in 1/1024s the indication is shown.
			7	request light pattern	Request the current light pattern. The desired light type is sent as an uint8 with preceding Op Code 0x07. The response to this control point is 0x20 followed by the light type as uint8 and pattern as an array of uint8. See Table 3: Light type enumeration for light types and Table 4: Pattern description for pattern description.
			8	set light pattern	Initiate the procedure to change the light pattern. The desired light type (as uint8) and new pattern (as an array of uint8) is sent with preceding Op Code 0x08. The response to this control point is 0x20. See Table 3: Light type enumeration for light types and Table 4: Pattern description for pattern description.
			10	set plunger position	Initiate the procedure to set the plunger opening value. The requested opening value (in percent with a resolution of 0.5) is sent with preceding Op Code 0x09. The response to this control point is Op Code 0x20.
			11	request motor configuration	Request the current motor configuration. The response to this Op Code is 0x20 followed by the motor configuration. See Table 5: Motor configuration for configuration description.
			12	set motor configuration	Initiate the procedure to set the motor configuration. The new configuration is sent with preceding Op Code 0x0C. The response to this control point is 0x20. See Table 5: Motor configuration for configuration description.

			13	request motor sensor calibration	Request the motor sensor calibration data. The desired calibration type is sent with preceding Op Code 0x0D. The response to this control point is 0x20 followed by the calibration data. See Table 6: Motor sensor calibration data for sensor calibration data
			14	clear/start motor sensor calibration	Initiate the procedure to clear or start the desired motor sensor calibration. The desired calibration type is sent as uint8 with preceding Op Code 0x0E, followed by 0 to clear current data or 1 to start the calibration. The response to this control point is 0x20 followed by the new calibration data (in case of starting). See Table 6: Motor sensor calibration data for sensor calibration data
			15	request travel sensor data	Request the travel sensor data. The response to this control point is 0x20 followed by the sensor data data. See Table 7: Travel sensor data for sensor data
			16	clear/start motor sensor calibration	Initiate the procedure to clear or start the travel sensor calibration. To clear the sensor data send Op Code 0x10 without any data. To start the calibration send Op Code ßx10 followed with the seatpost travel in mm (as uint16). The response to this control point is 0x20 followed by the new sensor data (in case of starting). See Table 7: Travel sensor data for sensor data
			17	request indexed travel configuration	Request the indexed travel configuration. The response to this control point is 0x20 followed by the configuration. See Table 8: Indexed travel configuration
			18	set indexed travel configuration	Initiate the procedure to set the indexed travel configuration. The new configuration is sent with preceding Op Code 0x12. The response to this control point is 0x20. See Table 8: Indexed travel configuration
			32	Response Code	The response code is followed by the requested Op Code, the response value and optionally the response parameter

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>0-0</td><td>Reserved for future use</td><td></td></tr>
<tr><td>15-31</td><td>Reserved for future use</td><td></td></tr>
<tr><td>33-255</td><td>Reserved for future use</td><td></td></tr>
</table>
</td></tr>
<tr><td>Parameter Value</td><td>Optional</td><td>variable</td><td>Refer to the Op Code table above for additional information on the possible values for this filed</td></tr>
<tr><td>Request Op Code
Information:
The Request Op Code is a sub field of the Parameter Value for "Response Code" Op Code.
C1: This Field is Mandatory for "Response Code" Op Code, otherwise this field is Excluded.</td><td>C1</td><td>uint8</td><td>Refer to the Op Code table above for additional information on the possible values for this filed</td></tr>
<tr><td>Response Value
Information:
The Request Op Code is a sub field of the Parameter Value for "Response Code" Op Code.
C1: This Field is Mandatory for "Response Code" Op Code, otherwise this field is Excluded.</td><td>C1</td><td>uint8</td><td>Enumerations
<table>
<tr><th>Key</th><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Success</td><td>Response for successful operation.</td></tr>
<tr><td>2</td><td>Op Code not supported</td><td>Response if unsupported Op Code is received.</td></tr>
<tr><td>3</td><td>Invalid Parameter</td><td>Response if Parameter received does not meet the requirements of the service or is outside of the supported range</td></tr>
<tr><td>4</td><td>Operation Failed</td><td>Response if the requested procedure failed</td></tr>
<tr><td>0-0</td><td>Reserved for future use</td><td></td></tr>
<tr><td>5-255</td><td>Reserved for future use</td><td></td></tr>
</table>
</td></tr>
<tr><td>Response Parameter
Information:
The Response Parameter is a sub field of the Parameter Value for "Response Code" Op Code.
C2:This Field is Optional for "Response Code" Op Code, otherwise this field is Excluded.</td><td>C2</td><td>variable</td><td>Note: The Response Parameter Value of the response to the Control Point is a variable length field to allow a list of different values defined by the Service Specification</td></tr>
</table>

Key	Name	Additional Information
0	Dropper actuator	
16	16 LED rear light	
44	44 LED rear light	
1-15, 17-43, 45-255	Reserved for future use	

Table 1: Device usage enumeration

Name	Format	Additional Information
type	<table><tr><th>Key</th><th>Name</th></tr><tr><td>0</td><td>LCO</td></tr><tr><td>1</td><td>NMC</td></tr><tr><td>1-255</td><td>Reserved for future use</td></tr></table>	cell chemistry
design capacity **Information**: Unit is in Coulomb with a resolution of 1 **Unit**: org.bluetooth.unit.elctric_charge_coulomb **Exponent**: Decimal, 0	uint16	
actual capacity **Information**: Unit is in Coulomb with a resolution of 1 **Unit**: org.bluetooth.unit.elctric_charge_coulomb **Exponent**: Decimal, 0	uint16	This information is only mandatory when requesting. When setting the battery information this filed is excluded.

Table 2: Battery information

Key	Name	Additional Information
0	Front light	
1	Rear light	
2	Position light	
3	Brake indicator light	
4	Left indicator light	
5	Right indicator light	
6-255	Reserved for future use	

Table 3: Light type enumeration

Name	Format	Additional Information
pattern	array of uint8	<table><tr><td></td><td>bit 7</td><td>bit 6</td><td>bit 5</td><td>bit 4</td><td>bit 3</td><td>bit 2</td><td>bit 1</td><td>bit 0</td></tr><tr><td>pattern[0]</td><td>LED_8</td><td>LED_7</td><td>LED_6</td><td>LED_5</td><td>LED_4</td><td>LED_3</td><td>LED_2</td><td>LED_1</td></tr><tr><td>pattern[1]</td><td>LED_{16}</td><td>LED_{15}</td><td>LED_{14}</td><td>LED_{13}</td><td>LED_{12}</td><td>LED_{11}</td><td>LED_{10}</td><td>LED_9</td></tr><tr><td colspan="9">...</td></tr><tr><td>pattern[n]</td><td>LED_{8n}</td><td>LED_{8n-1}</td><td>LED_{8n-2}</td><td>LED_{8n-3}</td><td>LED_{8n-4}</td><td>LED_{8n-5}</td><td>LED_{8n-6}</td><td>LED_{8n-7}</td></tr></table>

Table 4: Pattern description

Name	Format	Additional Information
open timeout **Information**: Unit is in seconds with a resolution of 1/1024 **Unit**: org.bluetooth.unit.time.second **Exponent**: Binary, -10	uint16	The time the actuator motor will be activated to open. This timeout is only used if no motor sensor is available or not calibrated yet
close timeout **Information**: Unit is in seconds with a resolution of 1/1024 **Unit**: org.bluetooth.unit.time.second **Exponent**: Binary, -10	uint16	The time the actuator motor will be activated to close

Table 5: Motor configuration

Name	Format		Additional Information
calibration type	**Key**	**Name**	
	0	comp threshold	the comparator thresholds used for the sensor
	1	full open count	the number of counts until the plunger is in fully open position
	2-255	Reserved for future use	
upper comp threshold **Information**: Unit-less with a resolution of 1 **Exponent**: Decimal, 0 **Information:** This field is mandatory for “comp threshold”, otherwise excluded	uint8		If both, upper and lower thresholds are 0, sensor hasn’t been calibrated yet
lower comp threshold **Information**: Unit-less with a resolution of 1 **Exponent**: Decimal, 0 **Information:** This field is mandatory for “comp threshold”, otherwise excluded	uint8		
full open count **Information**: Unit-less with a resolution of 1 **Exponent**: Decimal, 0 **Information:** This field is mandatory for “fully open count”, otherwise excluded	uint16		

Table 6: Motor sensor calibration data

Name	Format	Additional Information
seatpost travel **Information**: Unit is in millimeters with a resolution of 1 **Unit**: org.bluetooth.unit.length.metre **Exponent**: Decimal, -3	uint16	
sensor offset **Information**: Unit is in micrometers with a resolution of 1 **Unit**: org.bluetooth.unit.length.metre **Exponent**: Decimal, -6	int32	

Table 7: Travel sensor data

Name	Format	Additional Information
opening blocking time **Information**: Unit is in seconds with a resolution of 1/1024 **Unit**: org.bluetooth.unit.time.second **Exponent**: Binary, -10	uint16	The time the opening position is ignored
closing timeout **Information**: Unit is in seconds with a resolution of 1/1024 **Unit**: org.bluetooth.unit.time.second **Exponent**: Binary, -10	uint16	The timeout to close when no indexed position is reached.
indexed positions **Information**: Unit is in millimeters with a resolution of 1 **Unit**: org.bluetooth.unit.length.metre **Exponent**: Decimal, -3	array of int16	

Table 8: Indexed travel configuration