

TWN4

BLE Protocol Specification

DocRev3, January 23, 2018



Elatec GmbH

Contents

1 Bluetooth Low Energy (BLE) Feature	3
1.1 Advertising	3
1.2 Connection Establishment	3
1.3 Generic Access Profile (GAP)	4
1.4 Attribute Protocol (ATT)	4
1.5 Generic Attribute Profile (GATT)	5
1.6 Appendix: BLE Technical Information	6

1 Bluetooth Low Energy (BLE) Feature

The traditional Bluetooth standard is convenient for constant-flow media transfer applications such as video streaming. The Bluetooth Low Energy standard was introduced for applications requiring a lower power consumption profile. Data is sent in bursts, followed by periods of electrical idle.

For information regarding the Bluetooth Low Energy Standard please see document "*Designing for Bluetooth Low Energy*"[1] from Silicon Labs.

The TWN4 can be configured as a GATT server, accepting requests for information from clients such as a Bluetooth-capable cellphone.

1.1 Advertising

Advertisement is one of the most fundamental operations in BLE wireless technology. A Bluetooth Low Energy device broadcasts packets on one more advertisement channels for remote devices to pick up. This action informs the potential link partners of the device's capabilities and willingness to interact.

The TWN4 BLE module supports advertisement on all 3 channels dedicated for this purpose (37, 38 and/or 39), with interval of transmission between 20ms and 10,240ms. It supports data payload.

TWN4 API allows the User to configure the following parameters:

- Advertisement interval (20ms to 10240ms)
- Advertisement channels (37, 38 and/or 39)
- Discoverability mode (Not discoverable, Generic Discoverable, Limited Discoverable, Broadcast)
- Connectability mode (Not Connectable, Directed Connectable, Undirected Connectable, Scannable)
- Payload Size (0 to 31 Byte)

1.2 Connection Establishment

BLE connections use CRCs, acknowledgments and retransmissions of lost data in order to ensure robust data exchanges. In addition, the Bluetooth Low Energy connections use Adaptive Frequency Hopping (AFH) to detect and adapt to the surrounding RF conditions and provide a reliable physical layer. Connections also support encryption and decryption of data to ensure its confidentiality.

A device must advertise its willingness to connect (within the packets it broadcasts), in order to initiate a BLE connection.

The application typically controls the following connection parameters:

- Minimum Connection Interval (7.5 ms)
- Maximum Connection Interval (4000 ms)

- Connection (slave) latency (0 to 500 (connection intervals))
- Supervision timeout (100 ms to 32000 ms)

1.3 Generic Access Profile (GAP)

The Generic Access Profile or GAP is one of the first layers every Bluetooth developer encounters. The GAP controls the details of a device's Visibility and Connectability with respect to other devices.

GAP defines device roles that provide specific requirements. Roles allow devices to have radios that either transmit (TX) only, receive (RX) only, or do both:

- Broadcaster (TX only): Sends advertising events and broadcast data.
- Observer (RX only): Listens for advertising events and broadcast data.
- Peripheral (RX and TX): Always slave; advertises and is connectable. Designed for a simple device using a single connection to another device in the Central role.
- Central (RX and TX): Always master, never advertises. Designed for a device that is in charge of initiating and managing multiple connections.

A device can support more than one role, but only one role can be adopted at a given time:

- Connectable: Can make a connection. State: Non-connectable, connectable.
- Discoverable: Can be discovered (is advertising). State: None, limited, general.
- Bondable: If connectable, will pair with connected device for a long-term connection. State: Non-bondable, bondable.

1.4 Attribute Protocol (ATT)

Bluetooth Low Energy profiles expose a state of a device. The state is exposed as one or more values called attributes. The protocol to access these attributes is called the Attribute Protocol (ATT). The ATT defines the communication between two devices playing the roles of server and client, respectively, on top of a dedicated L2CAP channel. The Attribute protocol defines two roles:

- Server: The device that stores the data as one or more attributes
- Client: The device that collects the information for one or more servers

The client can access the server's attributes by sending requests, which trigger response messages from the server. For greater efficiency, a server can also send two types of unsolicited messages to a client that contain 2 attributes: notifications (which are unconfirmed) and indications (which require the client to send a confirmation). A client may also send commands to the server in order to write attribute values. Request/response and indication/confirmation transactions follow a stop-and-wait scheme.

Attributes are arrays that can vary from 0 to 512 bytes. They can be of fixed or variable length.

Attributes also have a type described by a UUID (Universally Unique Identifier). Two types of UUIDs are used:

- Globally unique 16-bit UUID, defined in the characteristics specification (www.bluetooth.com)
- Manufacturer-specific 128-bit UUIDs, which can be generated online (www.uuidgenerator.net)

Attributes also have permissions, which can be:

- Readable / Not readable
- Writable / Not writable
- Readable and writable / Not readable and not writable

The attributes may also require:

- Authentication to read or write
- Authorization to read or write
- Encryption and pairing to read or write

1.5 Generic Attribute Profile (GATT)

Generic Attribute Profile (GATT) is built on top of the Attribute Protocol (ATT) and establishes common framework for the data transported and stored by the Attribute Protocol. GATT defines two roles: Server and Client.

The GATT server stores the data transported over the Attribute Protocol and accepts ATT requests from the GATT client. The GATT server on the other hand sends responses to requests and when configured, sends indication and notifications to the GATT client when events occur on the GATT server. GATT also specifies the format of data contained on the GATT server.

1.6 Appendix: BLE Technical Information

UUID (hex)	Attr. Handle	Name	R/W	Type	Description
1800		General Access Profile		Service	Service 1
2a00	7	Device name	R	Descriptor	xgatt_1800_2A00, fix "TWN4 BLE"
2a01	9	Appearance	R	Descriptor	xgatt_1800_2A01, fix 0000
180a		Device Information		Service	Service 2
2a29	12	Manufacturer Name	R	Descriptor	xgatt_180A_2A29, fix "OEM BLE"
2a24	14	Model Number	R	Descriptor	xgatt_180A_2A24, "TWN4 MultiTech"
2a25	16	Serial Number	R	Descriptor	xgatt_180A_2A25
2a26	18	Firmware Revision	R	Descriptor	xgatt_180A_2A26, f.e. "V1.04,16.05.2017"

Table 1.1: Read-only Bluetooth Standard-defined GATT Services

UUID Name	Attr. Handle	Name	R/W	Type	Description
UID1		Generic Attribute		Service	Service 3
UID2	21	SPP Data	Read, Notify, Write no response	Descriptor	xgatt_spp_data var. length=255, HEX data.
UID3	24	SP1 Data	Notify, Write no response	Descriptor	xgatt_sp1_data var. length=20, HEX data.
UID4	27	SP2 Data	Indicate, Read	Descriptor	xgatt_sp2_data var. length=20, HEX data.

Table 1.2: Custom GATT Services defined within Firmware

UUID Name	User UUID (hex)	Description
UID1	"5a44c004-4112-4274-880e-cd9b3daedf8e"	SPP Service
UID2	"43c29edf-2f0a-4c43-aa22-489d169ec752"	xgatt_spp_data
UID3	"a897339f-adf0-4a2b-a2ef-4a57512e6eb6"	xgatt_sp1_data
UID4	"71f1ae4d-e1d1-438c-b05d-2c2c16abaaa7"	xgatt_sp2_data

Table 1.3: Definition of Custom UUIDs