Terminal I/O Profile

Client Implementation Guide

Release r02

Table of contents

1 Introduction

Terminal I/O is a proprietary Bluetooth Low Energy GATT profile for bidirectional serial data communication and GPIO status information exchange developed by Stollmann.

Bluetooth Low Energy GATT communication is generally carried out between a server ("peripheral") and a client ("central").

With TerminalIO, the server role is typically implemented by a Stollmann embedded module (e.g. BlueMod+SR, BlueMod+S etc.) meanwhile the client role - if not taken over by an embedded module as well - is implemented on a smart device.

While the Bluetooth Low Energy GATT functionality is regularly implemented within the smart device's operating system / Bluetooth protocol stack, the TerminalIO client implementation must be provided by the application developer.

This document gives an overview over the TerminalIO client implementation tasks.

For App development under **Apple iOS®** and **Google Android®** Stollmann provides ready-to-use sample code of complete TerminalIO client implementations including sample Apps. Please contact Stollmann for further details.

2 TerminalIO Profile Overview

2.1 Generic Attribute Profile (GATT)

A Bluetooth Low Energy connection always consists of a GATT server and a GATT client. The GATT server stores data values, which are hierarchically structured in services and characteristics (see **Figure 1**), while the GATT client can read and write these values and can be notified by the server on value changes. Each Characteristic has its own properties and can have an optional Client Characteristic Configuration, which allows configuration of the characteristic (e.g. enable notifications). A short introduction to GATT is given in [1].

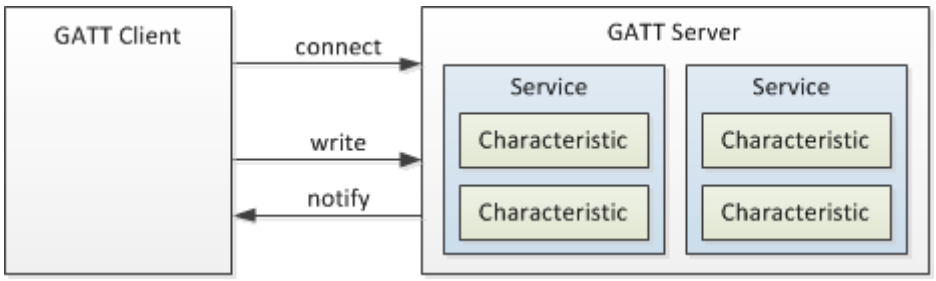

Figure 1 - Hierarchical structure of GATT.

2.2 GATT Structure of TerminalIO Profile

Figure 2 shows the GATT structure of Terminal I/O (optional elements grayed out) defined in [2].

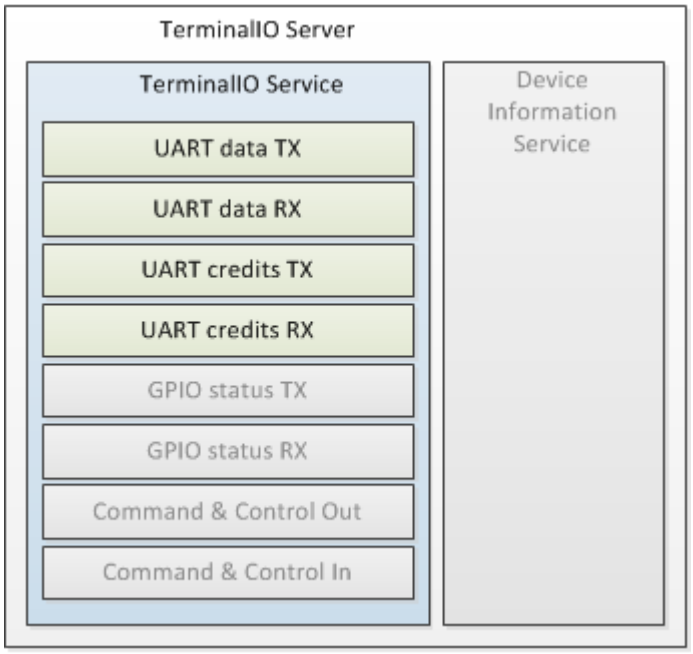

Figure 2 - GATT structure of TerminalIO Server.

The UART Data RX/TX characteristics are used for serial data exchange (see 6.2., 6.3).

The UART Credits RX/TX characteristics are used for serial data flow control (see 6.4., 6.5).

The (optional) GPIO status RX/TX characteristics are used for transmission of GPIO status information.

The (optional) Command & Control In/Out characteristics are used for command exchange.

In the following chapters, only the mandatory characteristics (UART data, UART credits) will be discussed.

3 TerminalIO Connection Setup

Figure 3 shows the TerminalIO connection setup procedure.

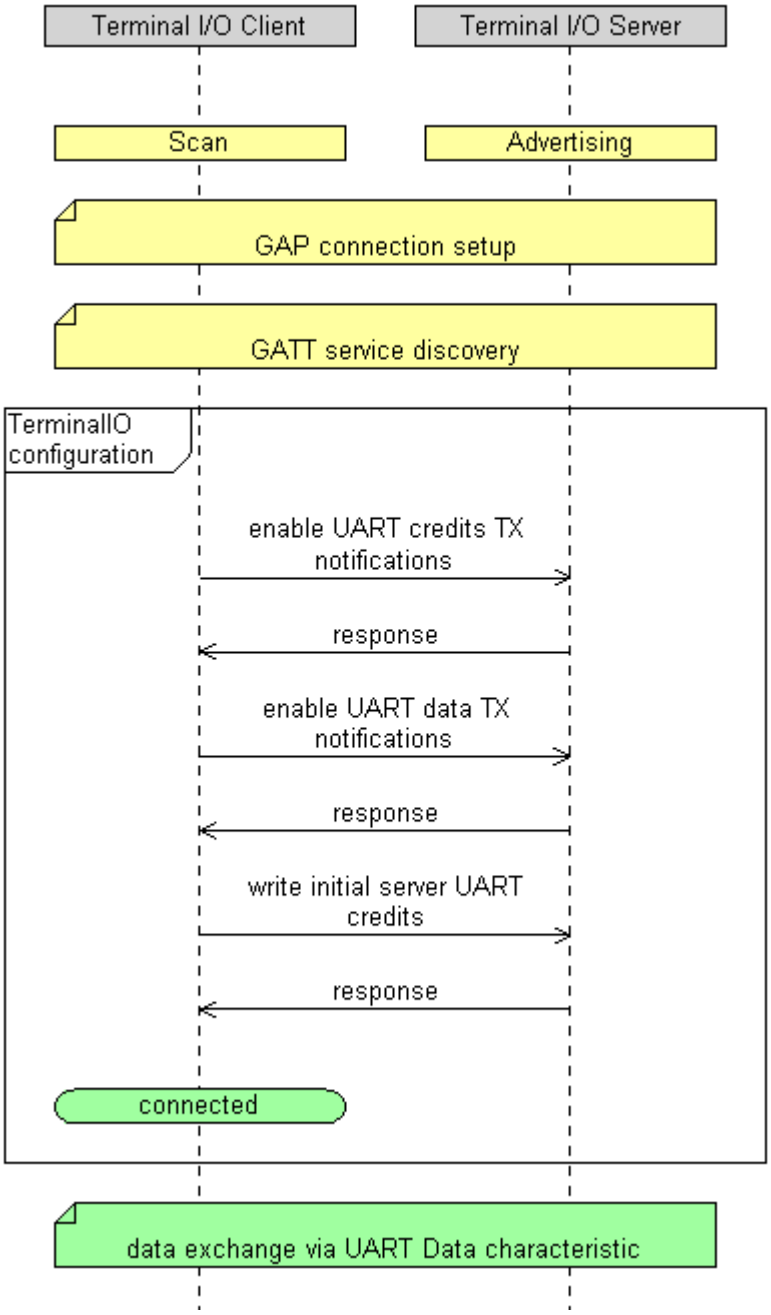

Figure 3 TerminalIO connection setup

In detail, the TerminalIO connection setup consists of the following steps:

- The TerminalIO client scans for Bluetooth Low Energy devices advertising the TerminalIO service.
- The TerminalIO client establishes a Bluetooth Low Energy GATT connection to a detected TerminalIO server.
- The TerminalIO client performs a service discovery on the TerminalIO server.
- For the retrieved TerminalIO service, the TerminalIO client performs a characteristics discovery.

- The TerminalIO client subscribes to indications of the UART credits TX characteristic (see 6.4).
- The TerminalIO client subscribes to notifications of the UART data TX characteristic (see 6.2).
- The TerminalIO client transmits initial UART credits to the server (see 6.5).
- Once the TerminalIO client has received the response for the transmitted UART credits, the TerminalIO connection is considered established and indications for the UART credits TX characteristic and notifications for the UART data characteristic shall be expected at any time.

The order of the connection setup sequence is mandatory. Especially, the subscription to the UART credits TX characteristic has to be completed **before** subscribing to the UART data TX characteristic.

It depends on the number of UART credits granted by the TerminalIO server whether the client may write UART data to the server (see 4.2).

4 TerminalIO UART Data Exchange

4.1 Receiving UART Data

The TerminalIO server uses the UART Data TX characteristic to send UART data to the TerminalIO client via notifications.

In order to receive UART data, the TerminalIO client has to entitle the TerminalIO server to send data by granting UART credits via writing the number of credits to be granted to the UART credits RX characteristic (see 6.5).
1 UART credit refers to 1 UART data notification regardless of how many bytes (1 - 20) it may contain.

It is the TerminalIO client's responsibility to grant an appropriate number of UART credits in order to ensure a smooth data flow and to avoid unnecessary data traffic on the UART credits RX characteristic.

4.2 Sending UART Data

The TerminalIO client uses the UART Data RX characteristic to send UART data to the TerminalIO server.

UART data are sent by writing 1 - 20 bytes to the characteristic's value without response (see 6.3).

The TerminalIO client shall send UART data only when UART credits granted by the TerminalIO server (peripheral) are available.
1 UART credit refers to 1 UART data notification regardless of how many bytes (1 - 20) it may contain.

The TerminalIO client receives a specific number of UART credits from the TerminalIO server via indications on the UART credits TX characteristic (see 6.4).

It is the TerminalIO client's responsibility to track the number of UART credits granted by the server (peripheral) by adding the number of received credits to a credit counter and decrementing the credit counter for each UART data packet written to the server. Once the credit counter reaches 0, the TerminalIO client shall not send any UART data until having received additional UART credits from the server.

5 TerminalIO Bonding and Security

For TerminalIO connections, security requirements are determined by the TerminalIO server (peripheral).

The TerminalIO client performs the connection setup described in 3. If the TerminalIO server is configured to require a secure link, an appropriate authentication will be requested from the client during the subscription to the UART credits TX indications.

The client side of the bonding procedure will be carried out by the operating system's Bluetooth protocol stack.

After successfully finishing the bonding procedure, the TerminalIO client shall proceed with the connection setup as described in 3.

6 TerminalIO GATT Structure in Detail

6.1 TerminalIO Service

UUID: 0xFEFB

This is the TerminalIO GATT service containing all TerminalIO characteristics.

6.1.1 TerminalIO Manufacturer Specific Advertisement Data

The TerminalIO server advertisements contain the following manufacturer specific data:

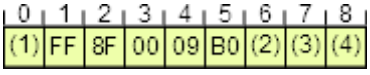

Figure 4 TerminalIO manufacturer specific data

Byte	ID	Description	Value
0	(1)	Length of manufacturer data	0x8
6	(2)	Advertising Compatibility Version	0x01
7	(3)	Terminal I/O server operation mode	see Table 2
8	(4)	Connection requested	0x01=true; 0x00=false

Table1 - TerminalIO manufacturer specific data

Value	Operating mode
0x00	bonding-only
0x01	functional
0x10	bondable and functional

Table 2 - TerminalIO operating mode values

6.2 UART Data TX Characteristic

UUID: 00000002-0000-1000-8000-008025000000

Type: uint8 array (20 bytes)

Properties: Notify

The TerminalIO client uses the UART Data TX characteristic to receive UART data from the server (peripheral) via notifications.

6.3 UART Data RX Characteristic

UUID: 00000001-0000-1000-8000-008025000000

Type: uint8 array (20 bytes)

Properties: Write without response

The TerminalIO client uses the UART Data RX characteristic to write UART data to the server (peripheral).

6.4 UART Credits TX Characteristic

UUID: 00000004-0000-1000-8000-008025000000

Type: uint8 (1 byte)

Properties: Indicate

The TerminalIO client uses the UART Credits TX characteristic to receive UART credits (0 - 255) from the server (peripheral) via indications.

6.5 UART Credits RX Characteristic

UUID: 00000003-0000-1000-8000-008025000000

Type: uint8 (1 byte)

Properties: Write

The TerminalIO client uses the UART Credits RX characteristic to grant UART credits (0 - 255) to the server (peripheral).

7 Related Documents

The *Terminal I/O Service Specification* [3] specifies the Terminal I/O Service and its characteristics.

The *Terminal I/O Profile Specification* [2] specifies the Terminal /IO Profile including server operating modes, server and client role requirements and connection, configuration and data exchange procedures.

8 References

[1] Bluetooth SIG, „GENERIC ATTRIBUTE PROFILE (GATT),“ 2013. [Online]. Available: https://developer.bluetooth.org/TechnologyOverview/Pages/GATT.aspx.

[2] Stollmann, Terminal I/O Profile.

[3] Stollmann, Terminal I/O Service.

9 History

Version	Release Date	By	Change description
r01d01	2013-09-05	kud	initial draft
r01d02	2013-09-17	kud	section *functional overview*; section *operation modes*; minor approvements
r01	2013-09-19	hb	First release version
r02_d01	2014-03-14	lv	Adoption of TerminalIO V2.1 Specification General revision
r02	2014-06-13	or	Second release version

Stollmann Entwicklungs- und Vertriebs-GmbH
Mendelssohnstraße 15 D
22761 Hamburg
Germany

Phone: +49 (0)40 890 88-0
Fax: +49 (0)40 890 88-444
E-mail: info@stollmann.de
www.stollmann.de