Bluetooth API

Updated over 6 months ago

Table of contents

This page describes the Bluetooth API that you can use to work with Atmotube directly via BLE connection (without Atmotube App). e.g. you can write custom program for Windows, Raspberry Pi, etc.

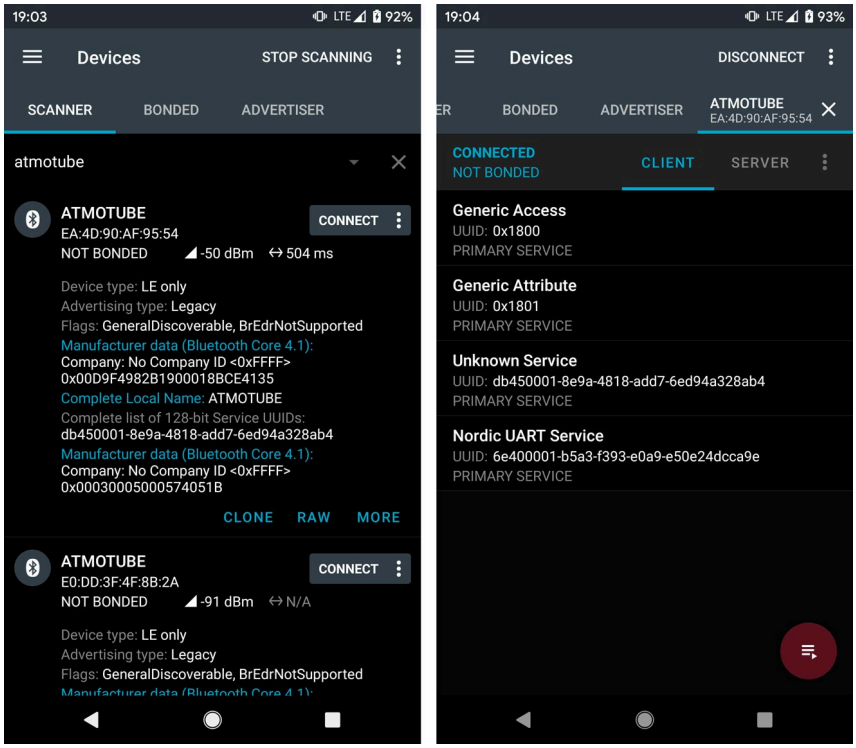
There are 2 ways to get data from an Atmotube device:

1. Bluetooth Broadcast
2. GATT characteristics

You can check the following code samples:

1. Atmotube Android library

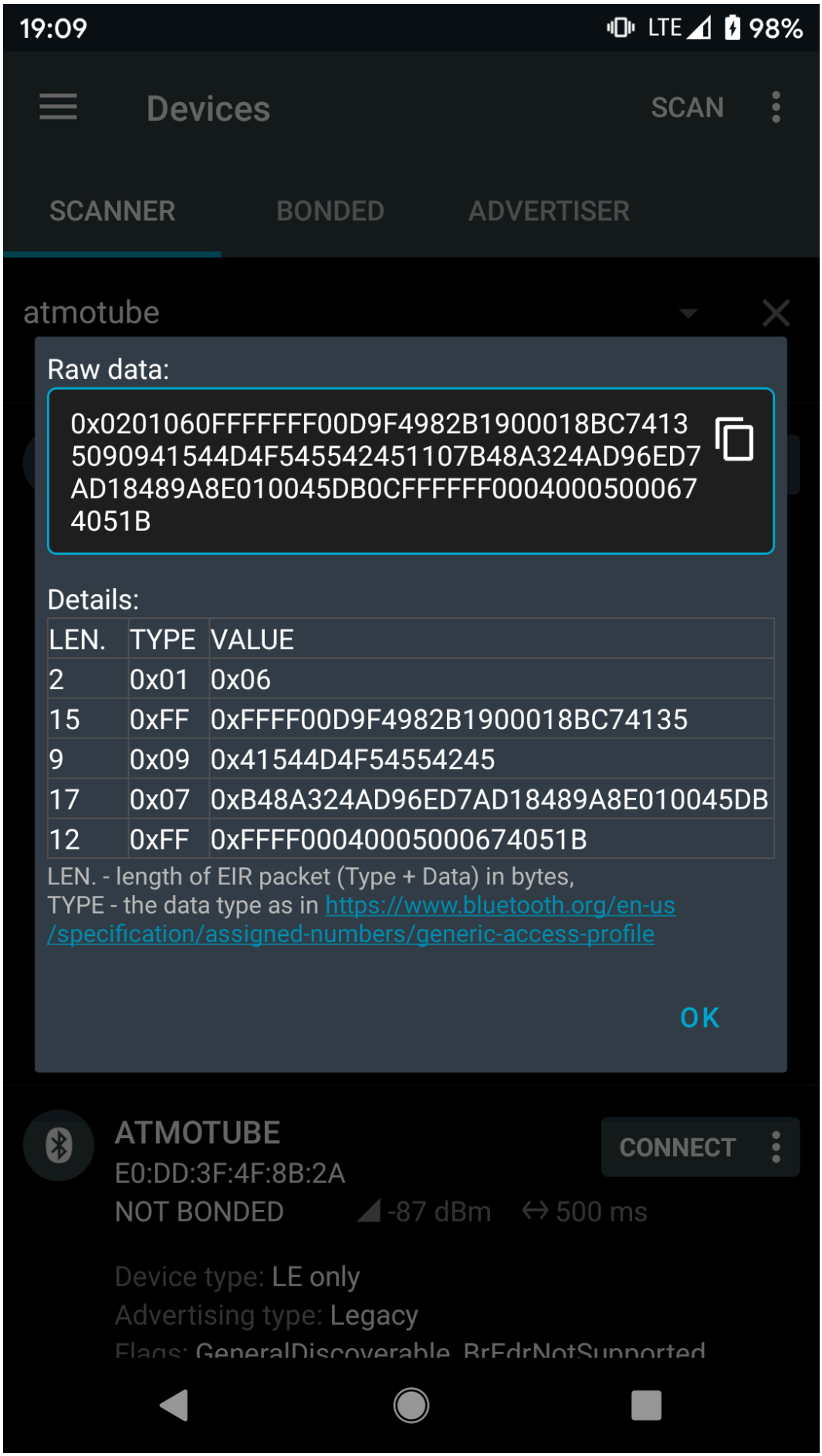
To analyze Bluetooth packets and check GATT characteristics, you can use the Nordic "nRF Connect" app for Android and iOS.



Note: This page describes the data format for the latest available firmware. Previous firmware versions may work in a different way.

Bluetooth Broadcast

To view a raw packet, press the RAW button.



Raw Atmotube PRO packet example:
`0x0201060FFFFFFF00D9F4982B1900018BC74135090941544D4F545542451107B48A324AD96ED7AD18489A8E010045DB0CFFFFFF0004000500067 4051B`

BLE Advertising Packet Format

Len	Type	Description	Value
2	`0x01`	Flags	`0x06`
15	`0xFF`	Manufacturer Specific Data	`0xFFFF00D9F4982B19 00018BC74135` **Company Identifier** • **Size**: 2 bytes • **Example**: `0xFFFF` **SGPC3 VOC, ppb** • **Size**: 2 bytes • **Example**: `0x00D9` **Device ID** • **Size**: 2 bytes • **Example**: `0xF498` **BME280 humidity, %** • **Size**: 1 byte • **Example**: `0x2B` • **Description**: BME280 humidity, % **BME280 temperature, °C** • **Size**: 1 byte • **Example**: `0x19` **BME280 pressure, mbar * 100** • **Size**: 4 bytes • **Example**: `0x00018BC7` **Info Byte** • **Size**: 1 byte • **Example**: `0x41` **Battery Level, %** • **Size**: 1 byte • **Example**: `0x35`
9	`0x09`	Complete local name	`0x41544D4F54554245` **ATMOTUBE**
17	`0x07`	Complete List of 128-bit Service Class UUIDs	**Atmotube PRO** `0xB48A324AD96ED7AD`

			`18489A8E010045DB` DB450001-8E9A-4818-ADD7-6ED94A328AB4
12	`0xFF`	Manufacturer Specific Data (scan response)	`0xFFFF000400050006 74051B` **Company Identifier** • **Size**: 2 bytes • **Example**: `0xFFFF` **PM1, ug/m³** • **Size**: 2 bytes • **Example**: `0x0004` **PM2.5, ug/m³** • **Size**: 2 bytes • **Example**: `0x0005` **PM10, ug/m³** • **Size**: 2 bytes • **Example**: `0x0006` **FW Version Major** • **Size**: 1 byte • **Example**: `0x73` **FW Version Minor** • **Size**: 1 byte • **Example**: `0x03` **FW Version Build** • **Size**: 1 byte • **Example**: `0x02`

SGPC3 data format

Size, bytes	Example	Description
2	`0x0294`	VOC data 0294 (hex) -> 660 (int) ppb / 1000 = 0.66 ppm

BME280 data format

Size, bytes	Example	Description
1	`0x1E`	humidity 1E (hex) -> 30 (int) %
1	`0x1C`	temperature 1C (hex) -> 28 (int) °C
4	`0x000182E0`	pressure 182E0 (hex) -> 99040 (int) / 100 = 990.40 mbar

SPS30 data format

Size, bytes	Example	Desription
2	`0x0004`	PM1 -> 0004 (hex) -> 4 (int) ug/m³
2	`0x0005`	PM2.5 -> 0005 (hex) -> 5 (int) ug/m³
2	`0x0006`	PM10 -> 0006 (hex) -> 6 (int) ug/m³

Info byte data format

Bits	Description
MSB 8	Reserved
7	SGPC3 pre-heating status `0` - SGPC3 is pre-heating `1` - device is ready
6	Reserved
5	Device charging timer status `0` - USB power was connected more than 30 minutes ago `1` - USB power was disconnected less than 30 minutes ago
4	Device charging status `0` - device is not charging `1` - device is charging
3	Device bonding status `0` - device is not bonded `1` - device is bonded
2	Device error status `0` - no error `1` - error code
LSB 1	PRO only - PM sensor status `0` - PM sensor is OFF `1` - PM sensor is ON

GATT characteristics

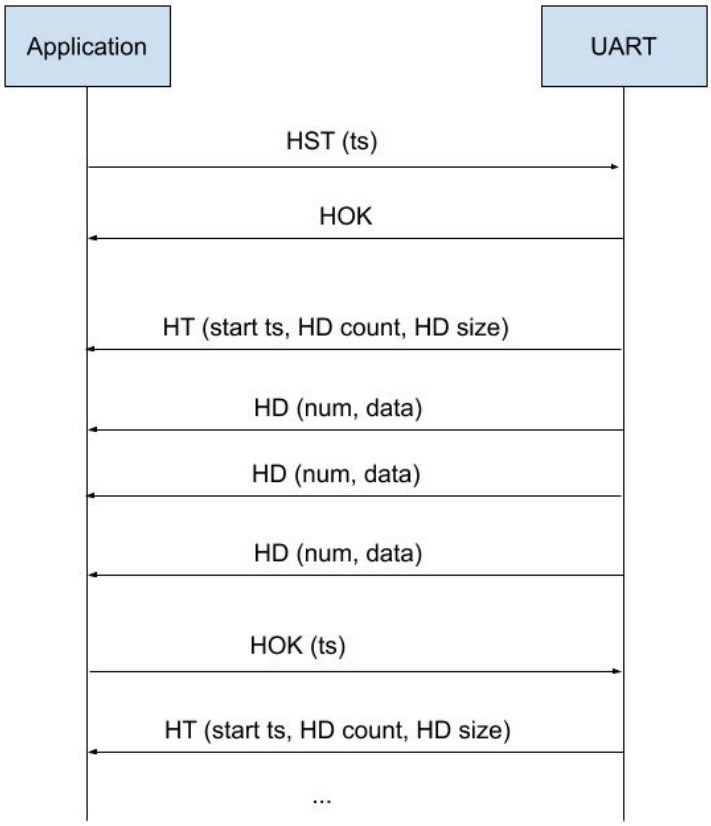
UUID	Description
6E400001-B5A3-F393-E0A9-E50E24DCCA9E	UART service
6E400002-B5A3-F393-E0A9-E50E24DCCA9E	UART RX characteristic
6E400003-B5A3-F393-E0A9-E50E24DCCA9E	UART TX characteristic

Atmotube PRO

UUID	Description
DB450001-8E9A-4818-ADD7-6ED94A328AB4	Atmotube PRO service
DB450002-8E9A-4818-ADD7-6ED94A328AB4	SGPC3 characteristic (4 bytes) - little-endian SGPC3 ppb, TVOC - 2 bytes SGPC3 (reserved) - 2 bytes
DB450003-8E9A-4818-ADD7-6ED94A328AB4	BME280 characteristic (8 bytes) - little-endian Humidity - 1 byte Temperature - 1 byte Pressure - 4 bytes Temperature - 2 bytes (extended precision, 0x0A0A (hex) -> 2570 (int) / 100 -> 25.7 °C) *For temperature, humidity, pressure conversion - see broadcast section.*
DB450004-8E9A-4818-ADD7-6ED94A328AB4	Status characteristic (2 bytes) - little-endian Info - 1 byte Battery level - 1 byte *For info and battery conversion - see broadcast section.*
DB450005-8E9A-4818-ADD7-6ED94A328AB4	PM characteristic (12 bytes) - little-endian PM1 - 3 bytes PM2.5 - 3 bytes PM10 - 3 bytes PM4 - 3 bytes 0x0003AD (hex) -> 941 (int) / 100 -> 9.41 ug/m³

Data History Protocol

History data sending is implemented via UART service. You have to send commands via RX and handle device answers via TX UART characteristics.



1. After UART connection, mobile device sends HST (history) request with current timestamp. UART service responds with HOK (history OK) acknowledge.

Command, ASCII, 3 bytes	Unix time, 4 bytes—current system time
HST	XXXX

Command, ASCII, 3 bytes
HOK

2. If the device has not synced history available, it starts data sending with HT packed following one or several HD packets. HT packet contains timestamp of the starting HD packet and number of HD packets and size. Mobile devices acknowledge HD packets received by sending the HOK command with the current timestamp.

Command, ASCII, 2 bytes	1 byte	Unix time, 4 bytes—first HD packet time	Number of HD packets, 1 byte	Size of the one measurement in bytes, 1 byte
HT	0	XXXX	X	X

Command, ASCII, 2 bytes	1 byte	Packet number, 1 byte	Measurement 1	-	Measurement N
HD	0	X	XXXX	-	XXXX

Command, ASCII, 2 bytes	1 byte	Packet number, 1 byte	Measurement N+1	-	Measurement N+1+x
HD	0	X	XXXX	-	XXXX

Measurement

N	Size in bytes, 1 byte	Description
0	1	Temperature, °C
1	1	Humidity, %
2	2 - little-endian	VOC (ppb)
3	4 - little-endian	Pressure (mbar * 100)
4	2 - little-endian	PM1, ug/m³
5	2 - little-endian	PM2.5, ug/m³
6	2 - little-endian	PM10, ug/m³

If a mobile device receives HT + all HD packets, it acknowledges data by sending the HOK command.

Command, ASCII, 3 bytes	Unix time, 4 bytes—current system time
HOK	XXXX

Related Articles

How do I access and use the Atmotube Cloud API? ›

Bluetooth API Specification ›