BLE custom profile for B2 boards

Intro

This page describes the BLE profile (services and characteristics) as exposed by the B2 boards.

Table of contents

List of sdvertisments

- Multi-value Advertisment

Advertisment Details

Multi-value Advertisment

The multi-value advertisment is a manufacturer specific advertisment which relies on manufacturer specific data. This advertisment uses the UUID **AB3A94DD-B5D8-9996-7447-B5DA5738FD9D** and carries 4 byte user specific data per advertisment. Each advertisment has a type which determines the content of the following 30 bits. The general structure is:

Following is the content for the four possible types:

Type	Global meaning	Bit (s) LSB to MSB	Content	Comment
0	This type represents the speed and cadence values from the last calculation. The values correspond to the BT running speed and cadence profile.	11 bit	Speed	The running speed in m/s with a resolution of 1/256 s. To get a usable speed in cm/s the value needs to be multiplied by 400 and divided by 1024.
		8 bit	Cadence	The cadence in 1/minute.
		11 bit	Stride Length	The stride length in centimeter.
1	This type represents the total distance.	30 bit	Total distance	The total distance in decimeter.
2	This type represents the foot kinematics values from the last calculation.	10 bit	Pronation	The pronation angle in 1/10 degree.
		10 bit	Footstrike	The foot strike angle in 1/10 degree.
		10 bit	Range of Motion	The range of motion in 1/10 degree.
3	This type represents aditional device information.	1 bit	Download Ready	The device has data and is ready to diownload those Data, e.g. no workout is active.

		21 bit	Step count	The step count in steps.
		8 bit	Battery voltage	The voltage offset of the power supply in 1 /100 Volt. The base to which the offset must be applied is 1500 milli volt. Taking the value multiply it by 10 and adding the 1500 base will give the battery voltage in milli volt.

The advertisment uses a dynamic scheme as described in the firmware feature list (here). It can be seen in general as a round robin scheme over the four advertisment types with the folowing rules:

- The device issues one advertisment every second
- Only values that changed will be advertised.
- Unchanged values will be advertised 5 times before this advertisment type is omitted.
- If no changed values are available the type 4 advertisment is sent regardless of changed values

With that scheme it is guaranteed that a device in central role sees one advertisment per second and gets one update of the values after a maximum of 4 advertisments. It might get earlier updates if less values changed. Hence it is possible to see a sequence of adversiment types like 0-1-2-3-0-1-2-3 if all values changed or, e.g. 3-1-3-1-3-1-2-3-1-2 if first only the total distance changed and than new foot kinematics were available.

List of services

- Device Information Service
- Battery Service
- Running Speed And Cadence Service
- Step Count Service adidas
- Foot Kinematics Service adidas
- Data Stream Service adidas
- Device Control Service adidas

Service Details

Device Information Service

This is a standard service and follows the Bluetooth SIG specification.

Battery Service

This is a standard service and follows the Bluetooth SIG specification.

Running Speed And Cadence Service

This is a standard service and follows the Bluetooth SIG specification.

Step Counter Service adidas

This is a custom service identified by the UUID **A13A94E0-B5D8-9996-7447-B5DA5738FD9D** with the following content:

Characteristic UUID	Property	Type	Content
A13A94E1-B5D8-9996-7447-B5DA5738FD9D	Read	32 Bit unsigned Integer	The step count value

A13A94E2-B5D8-9996-7447-B5DA5738FD9D	Read	32 Bit unsigned Integer	The liftime total distance (since the last reset) in decimeter

Foot Kinematics Service adidas

This is a custom service identified by the UUID **A13A9417-B5D8-9996-7447-B5DA5738FD9D** with the following content:

Characteristic UUID	Property	Type	Content
A13A94F7-B5D8-9996-7447-B5DA5738FD9D	Read	32 Bit Float	The foot strike angle
		32 Bit Float	The range of motion in pronation
		32 Bit Float	The maximum pronation velocity
		32 Bit Float	The maximum pronation angle

Data Stream Service adidas

This is a custom service identified by the UUID **A13A94D0-B5D8-9996-7447-B5DA5738FD9D** with the following content:

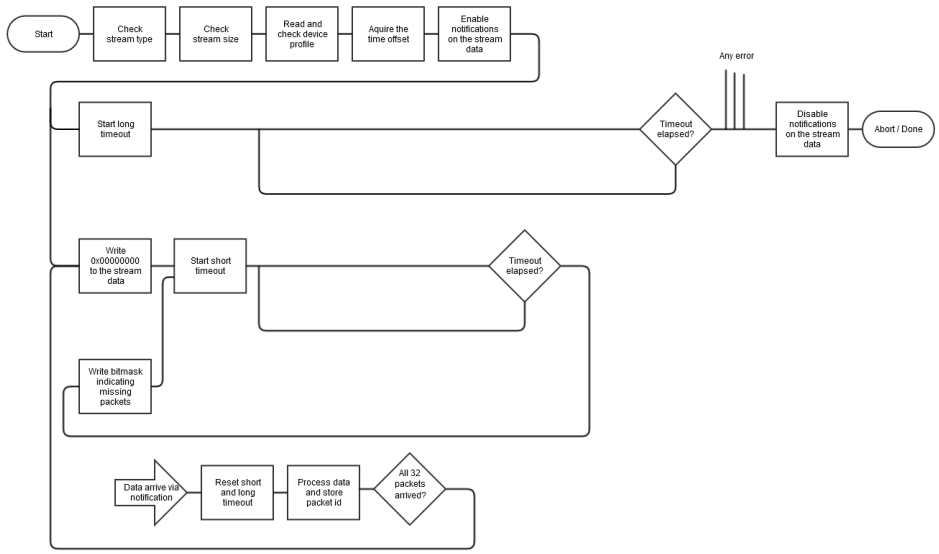
Characteristic UUID	Property	Type	Content
A13A94D1-B5D8-9996-7447-B5DA5738FD9D	Notify Write	Binary	The stream data as array of bytes. The first byte contains the packet id, the remaining bytes are the data. A 4 byte bit mask indicating missing packets (1..32) or 0 (zero) for 'no missing packets, start the next stream part'.
A13A94D2-B5D8-9996-7447-B5DA5738FD9D	Read	32 Bit unsigned Integer	The size of the stream in bytes.
A13A94D3-B5D8-9996-7447-B5DA5738FD9D	Read	8 Bit unsigned Integer	The stream type. Known types are: • 1- stream type 1 adidas

Usage of the stream to download data:

The stream download happens in junks of 32 packets. All stream data is received via notification. Hence reading the stream data characteristic will give the last packet sent to the listener but will not yield new data. Every 32 packets the device stops the notification process and waits for an acknowledgement from the listener. With the acknowledgement the listener has the chance to indicate missing packets. Since the device uses no timeouts for this wait step it is the responsibility of the listener to detect the right moment when packets are missing and the device should be notified.

> **Important**
>
> The device will not send stream data if a workout is still active. Hence the 'workout active bit' within the device profile should always be checked prior to a download.

In general you should follow this sequence:



It is advisable to re-read the steam size from time to time as an indication about remaining data. However, since stream download and active workout require exclusive use of the local B2 storage memory and stream download has the lower priority a download can be 'aborted' by the device at any time without notice. In such a case the device will just no longer send data via notifications to the listeners and listeners need to discover this condition via a timout. This would be the long timeout in the sequence diagram. However, the same thing would happen, if the device goes out of radio range and does not com back. Eventually the BT stack will give a disconnect but waiting for this might be too long.

In a normal process no packets will get lost and the next 32 packet set is started by acknowledging with a 0x00000000. If packets got lost the listener indicates missing packets with a bit mask. In this bit mask mising packets are indicaded by a set bit, e.g. if packet 2 has been missing the bit mask would be 0x00000004, if packet 10 and packet 2 are missing the bit mask would be 0x00000404. According to the bit mask the device will send the lost packets again via notification. This cloud be either 1 or 2 or all. In any case, the listener needs to check if all 32 packets have been received or continue the missing packet indication until all 32 packets have ben received successfully. Once the set is complete the listener starts the next 32 packet set by acknowledging the former set with a 0x00000000.

Although it might be advisable the keep an eye on the device profile during the download this can easily be omitted as the device will no longer send any data and then the long timeout would abort the download process.

With the knowledge of the stream size as determined at the beginning of a download process the listener could accumulate the already received packet sizes and determine the download end for itself. To reduce errors it might be better to just continue the download process and wait for the device no longer responding with new data, that is, the long timout elapses. Therefore the above diagram marks the end as either Abort or End.

> **Hint**
>
> For robustness it might be advisable to add repeats the the different crucial process steps, e.g. reading the stream type, before aboorting the download due to some transient problem.

Stream Types

From the stream type the content of the stream data can be determined. So far only the stream of type 1 is defined.

Stream Type 1 Data

The data of stream type 1 contains one data record per packet. Each packet follows the following identical structure:

Not all 8 possible record types are defined by may be in the future. The timestamp can be understood as an offset to the time difference as aquired at the beginning of the download process. Hence it cannot be used in itself but needs to be combined with this offset to get a timestamp with absolut time. See the following section about the device time and abolut time for a better understanding.

Record type	Global meaning	Bits (MSB to LSB)	Content
0	Accelerometer data	16 bit	Accelerometer data of the X axis in mG.
		16 bit	Accelerometer data of the Y axis in mG.
		16 bit	Accelerometer data of the Z axis in mG.
1	Gyroscope data	16 bit	Gyroscope data of the X axis in mDeg/s.
		16 bit	Gyroscope data of the Y axis in mDeg/s.
		16 bit	Gyroscope data of the Z axis in mDeg/s.
2	Magnetometer data	16 bit	Magnetometer data of the X axis in ?.
		16 bit	Magnetometer data of the Y axis in ?.
		16 bit	Magnetometer data of the Z axis in ?.
3	Speed and Cadence data	16 bit	Speed in m/s with a resolution of 1/256 s. To get a usable speed in cm/s the value needs to be multiplied by 400 and divided by 1024
		8 bit	Cadence in 1/minute.
4	Battery voltage	16 bit	The battery voltage in milli volt.
5	Foot Kinematics data	16 bit	The pronation angle in 1/10 degree.
		16 bit	The foot strike angle in 1/10 degree.
		16 bit	The range of motion in 1/10 degree.

A basic implementation of this stream download can be found in the source code repository of the RunBuddy reference app at *StreamingExecutorInertialDataStreamAdidas.java*.

Device time and Absolut time

As the device has no real time clock on board and due to the energy harvesting it could not be guaranteed that the real time clock would operate continiously, the device measures all times in device time. It is guaranteed that the device time only advances forward starting at zero. This cannot be guaranteed if a device firmware is newly flashed to the device. This is no problem if the the user followas the rule of downloading all data before a new firmware is flashed. Otherwise the old data might either be lost or downloaded with a largely wrong timestamp.

As the device has no other time than its local relative time all data are stored with a timestamp from this relative time. To map this timestamp the an absolut time the time difference between device relative time and absolut time needs to be determined prior to downloading the data. This can be done via the Device Control Service and the Device Time Characteristic. At some point in the download process the downloader needs to determine the current time offset between it's local time (or an UTC time if preferred) and the device time. This can be done by determining the absolut milliseconds at this specific moment, keeping this value and writing this value to the device. The device will calculate the offset from this absolut time to its relative time and makes this offset available via the Device Time Characteristic. Reading the offset back and combining it with the value written to the device the difference between device relative time and absolut time can be calculated and later applied to any timestamp of the downloaded data.

As it should never be assumed that the device is always powered and hence does not lose track of time it is recommended to recalculate the time offset prior to any stream download.

Reading the time offset without a prior written absolut time value will return a 0 offset. However, this behaviour should not be used to determine a 'power failure' of the device as a device might goes into deep sleep at any time stopping its time keeping but may got woken up by an incoming step before a power failure.

> **Important**
>
> Although the device advances the time when it has power it also fails to advance the time when no power is available. The time between a drained battery (power loss) and wakeup due to available energy cannot be determined by the device. If a power loss happens for the device at 3.600.000 milli seconds (1 hour) and the energy comes back after 10 minutes the realtive tiem of the device will still be at 3.600.000. The ten minutes are lost and gathered data appear as being gathered continuously.

Example:

Assume the device local time is at 3.600.000 and counting (one hour runtime under power). The downloader has a UTC time of 1.489.425.964.409 (13th March 2017, 17:26:04). If the downloader writes this time to the device it will read back an offset of 1.489.422.364.409. The device has its first sample stored at device time 1.800.000 (30 minutes back from now). The stream download of this first sample will give a timestamp of 1.800.000 combined with the offset the absolut time stamp is 1.489.424.164.409 (13th March 2017, 16:56:04).

Device Control Service adidas

This is a custom service identified by the UUID **A13A94F0-B5D8-9996-7447-B5DA5738FD9D** with the following content:

Characteristic UUID	Property	Type	Content
A13A94F1-B5D8-9996-7447-B5DA5738FD9D	Write	32 Bit unsigned Integer	A device control command. Known commands are: • Reset Step Count (MSB=3, all other bits are ignored) • Reset Total Distance (MSB=4, all other bits are ignored) • Clear Flash (MSB=5, all other bits are ignored) • Factory Reset (MSB=6, all other bits are ignored) • User Setup (MSB=7, next byte is gender (0=male, 1=female), next byte is body height in centimeter) • Callibration (MSB=9, the next byte is interpreted as 8 bit integer representing the calibration step (0, 1..9). Use 255 to explicitly abort the calibration sequence. Use 254 to stop a timed calibration step - e.g. gyroscope)

A13A94F2-B5D8-9996-7447-B5DA5738FD9D	Read Write	Binary Binary	8 byte binary data to be interpreted as 64 bit integer, representing the time offset between the device relative time and the given local time. 8 byte binary data to be interpreted as 64 bit integer, representing the time current local time. Usage: in order to determine the correct itme stamp of sored data the downloading device needs to know the offset between device relative time stamp and its local time as all data on the device are stored with device relative time stamp only. The device takes care that the relative time only advances and has no backward jumps. Hence a downloding device needs to write its local time to the device and read back the time offset. On every time stamp received from the device during the download the read offset needs to ba added in order to get the absolut time of the measurement.
A13A94F4-B5D8-9996-7447-B5DA5738FD9D	Read Write	16 Bit unsigned Integer	A 16 bit flags field which represents the current measurement profile of the device. The profile is not persisted on the device. Hence once the energy is drained and the device is powered off, the measurement profile will be reset to normal mode. The flags field contains the following values: • Bit 15: 0 = normal mode, 1 = extended mode • Bit 14: 0 = no workout active, 1 = workout active • Bit 13: 0 = no data download possible, 1 = data download possible • Bit 12: 0 = no valid body setup, 1 = height and gender set • Bit 11: 0 = no valid calibration, 1 = device calibrated • Bit 10..8: 1 = normal data aquisition algorithm, 2 = data aquisition as often as possible, 4 = continous data aquisition • Bit 6: 0 = do not store foot kinematics results, 1 = store foot kinematics results • Bit 5: 0 = do not store speed and cadence results, 1 = store speed and cadence results • Bit 4: 0 = do not store battery voltage, 1 = store battery voltage • Bit 2: 0 = do not store magnetometer raw data, 1 = store magnetometer raw data • Bit 1: 0 = do not store gyroscope raw data, 1 = store gyroscope raw data • Bit 0: 0 = do not store accelerometer raw data, 1 = store accelerometer raw data Normal mode without active workout represents a value of 0x0140 The measurement switch happens as soon as the new value is written, e.g. if a extended measurement profile with continous data aquisition is set, the device starts to store the raw sensor data and/or calculated results to local memory. Hence the measurement profile can be used to start /stop control the data aquisition.
A13A94F3-B5D8-9996-7447-B5DA5738FD9D	Read Write	Binary	An 11 byte binary value representing the 9D Inertial Sensor Configuration. The values contained are (MSB to LSB): • highest 2 byte: interpreted as 16 bit unsigned integer representing the sensitivity of the accelerator in g, that is 2=+/-2g • next 2 bytes: interpreted as 16 bit unsigned integer representing the sampling rate of the accelerator in 1/10 Hz, that is 10=1Hz • next 2 bytes: interpreted as 16 bit unsigned integer representing the sensitivity of the gyroscope in °/s, that is 125=+/-125°/s • next 2 bytes: interpreted as 16 bit unsigned integer representing the sampling rate of the gyroscope in 1/10Hz, that is 5000=500Hz • next 2 byte: interpreted as 16 bit unsigned integer representing the sampling rate of the magnetometer in 1/10 Hz, that is 250=25Hz • lowest byte: interpreted as 8 bit unsigned integer representing the measurement mode of the magnetometer (0 = 16 bit low noise, 1 = 16 bit, 2 = 14 bit, 3 = 12 bit) The sensor configuration is persisted by the device and survives over energy loses. Hence changing the sensor configuration will also have an influence on the normal mode data aquisition and calculation as the sensor data is directly fed into the algorithm.