FE-C over BLE

Simple how-to for engineers

17/02/2015

Introduction

This document will explain how to use the FE-C over BLE feature in Tacx Smart Trainers. Tacx designed this feature because Bluetooth is lacking open standard for trainers on Bluetooth.

It uses the ANT+ FE-C definition for the data content, but transports this data thru a BLE serial port service. Therefore a prerequisite for this technic is knowledge of the ANT+ FE-C protocol. A complete description of FE-C can be found at: http://www.thisisant.com/resources/fitness-equipment-device/

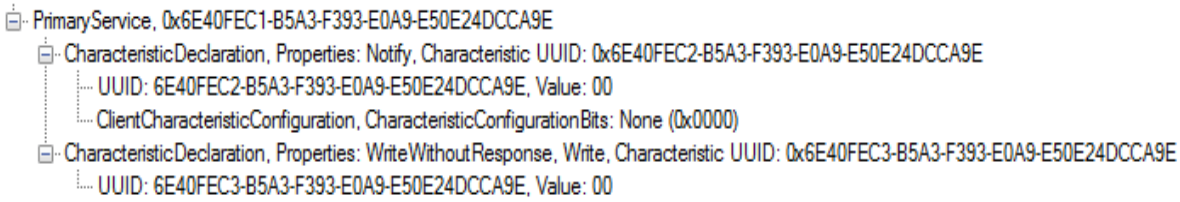
Revision	Effective Date	Author	Description
1.0.0	20-8-2015	D. de Nijs	Initial release.

Nordic Serialport Service

The transport method that is used is derived from the NordicUartService. This serial port service can basically send and receive arrays of data ranging from 1..20 bytes.

The only thing that is different is the service UUID and the two characteristics UUID's these are renamed to the Tacx standard UUID with the specific part for the FEC service starting 0xFEC…

In the figure below the UUID's are shown in there hierarchy order.



The 6E40-FEC2-B5A3-F393-E0A9-E50E24DCCA9E is the RX part of the service this characteristic will notify is the trainer sends out an packet(Notification must be turned on in the CharacteristicConfigurationBits).

Characteristic 6E40-FEC2-B5A3-F393-E0A9-E50E24DCCA9E is the TX part of the service, message that need to be send to the trainer can be written to this characteristic. The Ant+ FE-C wound exceed the 20 byte maximum , so there is no need to split message up in multiple sections.

The recommended rate of transmitting messages to the trainer is 4Hz since this is also the native rate over ANT. The trainer will also transmit messages with this rate.

Datacontent FE-C

The data that is generated by the trainer, and the data that is send to the trainer consists of a complete ANT message.

An example of a message from the trainer can look like this:

A4094E05101904000000FF30

When the data broken up into hex values and interpreted as an ANT message it looks like this:

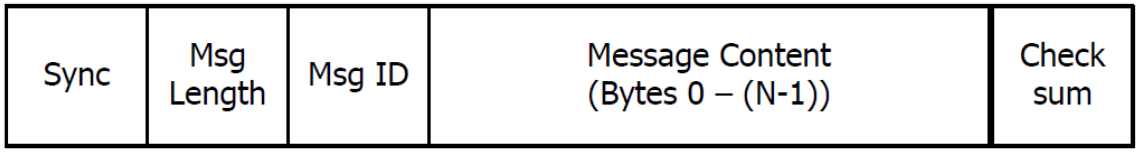


Figure 7-1. ANT serial message structure

As shown above, each message begins with a SYNC byte and ends with a CHECKSUM. The bytes are sent LSB first. Table 7-1 describes each component of the serial message shown above.

Table 7-1. ANT serial message components

Byte #	Name	Length	Description
0	SYNC	1 Byte	Fixed value of 10100100 or 10100101 (MSB:LSB) (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	MSG LENGTH	1 Byte	Number of data bytes in the message. (Refer to section 9.3)
2	MSG ID	1 Byte	Data Type Identifier 0: Invalid 1..255: Valid Data Type (See section 9 for details)
3..N+2	MESSAGE CONTENT	N Bytes	Content of the message as described in section 9.
N+3	CHECKSUM	1 Byte	XOR of all previous bytes including the SYNC byte

Section from document: ANT_Message_Protocol_and_Usage_Rev5.0.pdf

Example:

0xA4	0x09	0x4E	0x05	0x10 0x19 0x04 0x00 0x00 0x00 0xFF 0x24	0x30
Sync	Length	Type	Channel	Data	Checksum

This message is a General FE Data page

The message will always start with 0xA4 this is the ANT sync character, followed by the length of the data + 1. The type can vary i.e acknowledged, broadcast.

The channel number is our case channel 5 but the channel number is ignored when a message is send to the trainer.

The channel number is followed by (length-1) data bytes, the message end with a checksum which is the sum of all bytes excluding the 0xA4 sync byte.