GLÜCK Industrie-Elektronik GmbH

Zeichnungsnummer:	505014.001.7	**Blatt: 3**		**Rev.:**	*V1.7*
Zeichnungstitel:	***BLE UART Protocol***			**Datum:**	*25.01.2021*
Bezeichnung:	*Bownce C4/C5-Sample*			**Geändert:**	*08.01.2025*
Halbfabrikats/ Fertiggeräte Nr.:	*505014.001*				
Firma:	*Bownce*				
Kundenartikel Nr./ Bez.:	*n.a.*				
Sachbearbeiter:	*ThG*				
Geprüft:	*S.H.*				

Zeichnungsnummer: 505014.001.7.B3;V1.7 Stand 08.01.2025

Änderungshistorie

Datum	Version	Art der Änderung
25.01.21	1.0	Erstellt
19.03.21	1.0	Publish V1.0
23.03.21	1.1	Added Byte [10]: transmission counter to 2.1 Data Frame
21.04.21 26.07.21	1.2	Data Frame for Anchor specified Data Frame for Ball extended (2.2)
17.11.21	1.3	Data Frame for Ball extended (Swing period)
29.11.21	1.4	Offline session handling implemented
07.10.2022	1.5	Calculation of rope-tension fixed
06.02.2023	1.6	Info Frame removed
8.1.2025	1.7	Configuration for Power-Off-Timeout added

Inhaltsverzeichnis

0 Abstract

This document describes the BLE UART Protocol used in bownce ball and anchor.

1 General format

1.1 General format of payload frames

From Ball/Anchor to APP or from APP to Ball/Anchor

Start of frame	Payload Byte 0	Payload Byte 1	Payload Byte 2	Payload Byte 3	Payload Byte 4	Payload Byte 5	….	Payload Byte 18	End of frame

Start of frame (SOF): depends on payload

End of frame (EOF) : 0x2C (‘)

1.2 General format of command frames

From APP to Ball/Anchor

Start of frame	Command Byte	End of frame

In case the command triggers action on the device that does not result in responding with data frames, the device responds with ACK or NACK

(Such commands may be used in integration testing and/or during EOL testing)

ACK Frame:

The ACK Frame is a single Byte containing **0x40 ("@")**

NACK Frame

The NACK Frame is a single Byte containing **0x41**

2 Specific frame format

2.1 Data frame Anchor

The Data frame is used to transmit data from anchor to app

SOF Byte[0] : 0x23 „#“

Payload Byte [1]: tension [0]
Payload Byte [2]: tension [1]
Payload Byte [3]: battery level [0]
Payload Byte [4]: transmission counter [0]

EOF Byte[5] : 0x27

Where

Rope-tension in [g] = (tension [0] * 256) + tension [1]

Battery level in 100mv Increments

2.2 Data frame Ball

The Data frame is used to transmit data from ball to app

SOF Byte[0] : 0x23 „#“

Payload Byte [1]: timestamp [0]
Payload Byte [2]: timestamp [1]
Payload Byte [3]: timestamp [2]
Payload Byte [4]: hitcounter [0]
Payload Byte [5]: hitcounter [1]
Payload Byte [6]: swing time [0]
Payload Byte [7]: swing time [1]
Payload Byte [8]: consecutive hits [0]
Payload Byte [9]: force [0]
Payload Byte [10]: battery level [0]
Payload Byte [11]: Accuracy [0]
Payload Byte [12]: Flags [0]
Payload Byte [13]: transmission counter [0]
EOF Byte[14] : 0x27

Where

Timestamp = (timestamp [0] * 65536) + (timestamp [1] * 256) + timestamp [2]

Hitcounter = (hitcounter [0] * 256) + hitcounter [1]

consecutive hits = (consecutive hits[0])

swing time = (swing time [0] * 256) + swing time [1] in milliseconds

Flags:

Bit 0 : 1 → cheating detected, 0 → no cheating detected

Bit 1: 1 → direct hit detected , 0 → indirect hit detected

Bit 2: 1 → swing time is valid, 0 → swing time is not valid

(Since we need at least one free swing to measure the swing time, swing time may be invalid in case there were no free swings during the session)

2.3 Info frame

The Info frame was removed since device informations are covered by the device information service.

~~The „info frame“ is used to transmit informational data such as SW Version number from ball and/or anchor tot he app. The info frame is not send periodically by the ball/anchor, but can be requested by the APP.~~

~~SOF Byte[0] : 0x24 „$“~~

~~Payload Byte [1]: TBD~~

~~Payload Byte [2]: TBD~~

~~Payload Byte [3]: TBD~~

~~…. TBD ….~~

~~Payload Byte [19]: TBD~~

~~EOF Byte[20] : 0x27~~

2.4 Upload frame

The upload frame is removed since functionality is covered by the generic access profile (GAP)

~~The „Upload frame“ can be used to by the APP to send data to the ball/anchor device e.g. the device name.~~

~~SOF Byte[0] : 0x7B „{“~~

~~Payload Byte [1]: payload ID (0x01 → Device Name following)~~

~~Payload Byte [2]:~~

~~...~~

~~Payload Byte [N]: last byte of payload~~

~~EOF Byte[N+1] : 0x27~~

~~The receiving device respondes with **ACK** or **NACK**~~

2.5 Offline bowncing

Not implemented in C4/C5 samples

2.5.1 Request offline session information

The offline session information frame can be used from the app to request the information about the number of stored offline sessions.

SOF Byte[0] : 0x7C „|“

Payload Byte [1]: 0x02

EOF Byte[3] : 0x27

The receiving device respondes with the offline info frame:

SOF Byte[0] : 0x25

Payload Byte [1]: number of stored offline bowncing sessions

EOF Byte[20] : 0x27

2.5.2 Request offline session

SOF Byte[0] : 0x7C „|“

Payload Byte [1]: 0x03

Payload Byte [2]: session id to transmit

EOF Byte[3] : 0x27

The receiving device respondes with the session frame:

SOF Byte[0] : 0x26
Payload Byte [1]: session ID
Payload Byte [2]: session ID
Payload Byte [3]: session ID
Payload Byte [4]: session ID
Payload Byte [5]: hitcounter [0]
Payload Byte [6]: hitcounter [1]
Payload Byte [7]: average accuracy
Payload Byte [8]: average hitpower
Payload Byte [9]: consecutive hits
Payload Byte [10]: min. accuracy
Payload Byte [11]: max. accuracy
Payload Byte [12]: min. hitpower
Payload Byte [13]: max. hitpower
Payload Byte [14]: swing time [0]
Payload Byte [15]: swing time [1]
EOF Byte[16] : 0x27

Where:
Hitcounter = (hitcounter [0] * 256) + hitcounter [1]
consecutive hits = (consecutive hits[0])
swing time = (swing time [0] * 256) + swing time [1] in milliseconds

2.5.3 Set Power-Off-Timeout

This command can be sent from the mobile device to Ball as well as Anchor

SOF Byte[0] : 0x38

Payload Byte [1]: Timeout bit 24-32

Payload Byte [2]: Timeout bit 16-23

Payload Byte [3]: Timeout bit 8-15

Payload Byte [4]: Timeout bit 0-7

EOF Byte[5] : 0x27

The timeout is defined as 32Bit unsigned Integer.

On sucessfull write operation, the device responds with ACK (0x40)

The limit of the Timeout is 30 seconds. The upper limit is 3600 seconds. There will be NO ACK from the Device if the timeout exceeds the valid range of 30 to 3600 Seconds.

Please notice: changing the power off Timeout will only take place after the next power cycle. That means the old timeout value will remain active until the timeout occurs!

This is necessary because in case the timeout is set to a smaller value than the actual value, the timer could already be above the new value. In such a case the device would switch off immediately without the chance to store the new value in a nonvolatile memory.

2.5.4 Read back Power-Off-Timeout and Device ID

SOF Byte[0] : 0x25

Respond:

SOF Byte[0] : 0x24

Payload Byte [1]: stored Timeout bit 24-32

Payload Byte [2]: stored Timeout bit 16-23

Payload Byte [3]: stored Timeout bit 8-15

Payload Byte [4]: stored Timeout low bit 0-7

Payload Byte [5]: current Timeout bit 24-32

Payload Byte [6]: current Timeout bit 16-23

Payload Byte [7]: current Timeout bit 8-15

Payload Byte [8]: current Timeout low bit 0-7

Payload Byte [9-16]: 64Bit Unique Device ID

EOF Byte[17] : 0x27

Stored Timeout is the timeout stored in nonvolatile memory. This timeout will take place after the next power cycle.

Current Timeout is the timeout which is active in the remaining power cycle.

To clarify: If you update the timeout value with the command described in 2.5.3 and you read back the value right after updating it (during the same power cycle) the “stored” value shows the value that was written to the flash memory right before and the “current” value shows the “old” value.

In all other cases both values are identically.

E.g. the timeout value is 180 sec. and you update this to 120 sec. then the first read back shows 120 for the “stored” timeout and 180 for the “current” timeout. Then reaching the 180 seconds the device goes to off mode. After waking up again and reading back the timeout value, you will read 120 seconds for “stored” and “current” and the device goes to off mode after the timer reaches 120 seconds.