DigiBall API

8/9/2026, Build 64

Manufacturer Data ID: 0x03DE

Manufacturer Data: Advertisement (24 bytes)

Value Name	Byte Index (Big Endian)	Data Type (Big Endian)	Array Size	Value Range	Description
`mac_addr_3lsb`	0:2	uint8	3	0:255	Three least significant bytes of the six-byte MAC address
`ball_type`	3[7:4]	uint8	1	0:15	Defines ball diameter (in inches) 0 = 2.250 (American pool) 1 = 2.420 (Carom) 2 = 2.420 (Carom yellow ball) 3 = 2.063 (Snooker) 4 = 2.000 (English pool) 5 = 2.688 (Russian pyramid) 6-15 = Undefined
`device_type`	3[3:0]	uint8	1	0:15	Always 1
`build_number`	4	uint8	1	0:255	Firmware build number
`vdd_level`	5	uint8	1	0:255	VDD level linearly scaled from 0 to 255. Multiply by 0.02 to convert to volts. Nominal is 3.0V.
`gyro_clipping`	6[7]	boolean	1	True:False	True if any of the three gyroscope axis values exceed dynamic range.
`shot_number`	6[6:0]	uint8	1	0:127	Increases everytime the DigiBall detects a hit. Wraps from 127 to 0.
`charging`	7[7:6]	uint8	1		Charging status: 0 = Not charging (normal mode) 1 = Charging in progress 2 = Charging error 3 = Charging complete
`ship_mode`	7[5]	boolean	1	True:False	True if being held in ship mode, false otherwise.
`high_g`	7[4]	boolean	1	True:False	Always 1
`data_compressed`	7[3]	boolean	1	True:False	Format of sensor data packets: False = Multiple packets in round robin True = Single compressed packet
`seconds_motionless`	7[2:0]:8	uint16	1	0:1023	Number of seconds that the DigiBall as remained motionless since it has come to rest.
`auto_detect_time`	9	uint8	1	0:255	Sample index where first detected collision has occured. Collision will be detected if accelerometer magnitude exceeds conf_auto_threshold and at least conf_auto_min samples have elapsed. Sample rate is 104Hz.
`accel_x`	10	int8	1	-128:127	Live value of accelerometer X axis (aligned with black dot on DigiBall). 1G = 8 (black dot facing straight upwards).
`tip_estimation`	11	uint8	1	0:255	Estimation of percent from center.

`impulse`	12	uint8	1	0:255	Proportional to cue strike impulse. Speed in miles per hour is very roughly estimated by 0.06*impulse.
`spin_horz_dps`	13:14	int16	1	-32768: 32767	Horizontal spin in degrees per second. Right spin positive, left spin negative. Unit: degrees per second.
`spin_vert_dps`	15:16	int16	1	-32768: 32767	Vertical spin in degrees per second. Top spin positive, bottom spin negative. Unit: degrees per second.
`data_type`	17[7]	boolean	1	0:1	Type of sensor data 0 = 3D gyroscope magnitude 1 = 3D accelerometer magnitude
`data_ready`	17[6]	boolean	1	True:False	True if data processing has occured.
`data_packets`	17[5:3]	uint8	1	0:7	Number of sensor data packets - 1
`data_index`	17[2:0]	uint8	1	0:7	Packet index for sensor data.
`data`	18:23	uint8	6	0:255	Sensor data values.

Manufacturer Data: Scan Response (24 bytes)

Name	Byte Index (Big Endian)	Data Type	Array Size	Value Range	Description
`data_type`	0[7]	boolean	1	0:1	Type of sensor data 0 = 3D gyroscope magnitude 1 = 3D accelerometer magnitude
`data_ready`	0[6]	boolean	1	True:False	True if data processing has occured.
`data_packets`	0[5:3]	uint8	1	0:7	Number of sensor data packets - 1
`data_index`	0[2:0]	uint8	1	0:7	Packet index for sensor data.
`data`	1:23	uint8	23	0:255	Sensor data values.

Sensor Data Format (29 bytes)

`data_compressed = true`

`data_index`	Number of Samples	Sample Rate (Hz)	data_type	data[0:28]	Full Scale
0	29	13	0	gyroscope mag	$256 = \sqrt{3}\cdot 4000$ dps
0	29	13	1	accelerometer mag	$256 = \sqrt{3}\cdot 16$ Gs

`data_compressed = false`

`data_index`	Number of Samples	Sample Rate	data_type	data[0:28]	Full Scale
`[0:data_packets]`	`29*(data_packets+1)`	`13*(data_packets+1)`	0	gyroscope mag	$256 = \sqrt{3}\cdot 4000$ dps
`[0:data_packets]`	`29*(data_packets+1)`	`13*(data_packets+1)`	1	accelerometer mag	$256 = \sqrt{3}\cdot 16$ Gs

Dedicated Radio and Multicast (36 bytes)

Name	Byte Index (Big Endian)	Data Type (Big Endian)	Array Size	Description
Frame header	0:1	uint8	2	First two bytes of header are always 0xAA 0x55
Full mac address	2:7	uint8	6	Full six-byte MAC address of NRLLC device
RSSI	8	int8	1	Measured RSSI of NRLLC device
Adv. Type	9	uint8	1	0 = Advertisement from NRLLC device 1 = Scan response from NRLLC device
Payload	10:33	uint8	24	Manufacturer data from NRLLC device
CRC16	34:35	uint16	2	CRC16 of bytes 0:33

DIGIBALL

DigiBall Overview

The DigiBall is a digital cue ball designed to transmit information wirelessly. The information contains details representing the spin imparted upon the ball when struck by a billiards cue, an estimation of how far from the center of the ball the tip touched the surface of the ball, time series data on the spin magnitude and acceleration magnitude of the ball during motion, an estimation of the time of first collision with an object ball or rail, device information and charging status.

During normal operation, the DigiBall transmits BLE advertisements every 75-100ms. The advertisements contain read-only data embedded in the manufacturer data with the manufacturer ID 0x03DE. A scan response to the advertisement provides more datatime series sensor data. The advertisements will continue until the device times out by remaining motionless for 300 seconds, or if placed into Ship Mode. The DigiBall will stay asleep indefinitely until motion is detected. If placed into Ship Mode, the DigiBall will only wake up once placed on the charging pad.

Charging is performed by placing the ball black-dot-down into the center of the provided wireless charging pad. The status of charging is indicated by the charging bits. Ship mode can be entered by rotating the ball so that the black dot is facing straight upwards and momentarily placing the charger up-side-down over the black dot. Success is indicated by the ship_mode bit.

Upon waking, a reset is performed. shot_number will start at 0 and seconds_motionless will begin to increment. When a hit is detected, shot_number increments, seconds_motionless will be reset to 0, and the data_ready bit will be set to false. After a fraction of a second, the data_ready bit will be set to true and the following values will be come valid:

- gyro_clipping
- auto_detect_time
- tip_estimation
- impulse
- spin_horz_dps
- spin_vert_dps
- sensor data (29 bytes total, contained in both the advertisement and scan response)

If data_compressed is true, then data_index will always equal zero. The total number of samples will always be 29, which is the combination of data from the advertisement and scan response. The sample period of each sample is (data_packets+1)/104Hz, with a maximum time scale of (7+1)*29/104Hz = 2.231 seconds. If transmission of accelerometer data has been enabled in the settings then data_type will alternate between 0 and 1 every 0.3 seconds, indicating if data represents either gyroscopic magnitude data (0) or accelerometer magnitude data (1).

If data_compressed is false, then data_index will increment every 0.3 seconds from 0 to data_packets, and will wrap around back to zero. The packets must be stitched together in order to acquire the full sample set, and there is no guarantee that the packets will arrive sequencially. The tradeoff between data_compressed is resolution vs. transmittion time.

spin_horz_dps, spin_vert_dps

spin_horz_dps and spin_vert_dps represent the Cartesian components of the axis of rotation vector measured at the moment of impact initiating the cue ball hit, in reference to the gravitational vector (orthogonal to the table surface). spin_horz_dps is the amount of left to right spin in degrees per second (right spin positive). spin_vert_dps is the amount of top spin in degrees per second (top spin positive). A shot hit exactly in the center of the ball will have close to zero horizontal or vertical instantaneous spin. A shot hit along the vertical axis of the ball will have close to zero horizontal instantaneous spin, and a shot hit along the horizontal axis of the ball will have close to zero vertical instantaneous spin.

These components can be converted in to polar coordinates by calculating the spin magnitude and vector angle:

```
gyro_fs = 4000; //degrees per second full scale
accel_fs = 16.0; //Gs full scale
```

$$\mathtt{spin\_mag\_dps} = \sqrt{\mathtt{spin\_horz\_dps}^2 + \mathtt{spin\_vert\_dps}^2}$$

```
spin_angle_deg = 180/PI * atan2(spin_horz_dps,spin_vert_dps); //0 degrees is 12:00 on the face of a clock
```

The maximum measureable spin is 5657 dps (15.716 rps).

Detecting a new shot and resetting the shot clock timer

It is advisable to keep a running local clock and syncing it to seconds_motionless rather than relying on real time updates from seconds_motionless alone. This will prevent clock jitter if advertisements are missed. The following is pseudo-code to help with this:

```
static uint16_t last_shot_number = -1;
static float local_shot_clock = 0;

//Set up a timer to call this code every time_increment
//For example, time_increment = 0.1 seconds.

void timer_callback() {
    if (shot_number != last_shot_number) {
        local_shot_clock = 0;
        if (seconds_motionless==0) {
            last_shot_number = shot_number;
            //New shot, reset timer
        }
    } else {
        if (seconds_motionless>0) {
            local_shot_clock += time_increment;
        }
    }

    if (local_shot_clock > some_value) {
        //Time expired!
    }
}
```

FLASH Settings

A limited number of settings can be changed through via a connection and attribute update. The device enters connection broadcasts by performing gyro twists along the X-axis. Device always resets after successful characteristic write.

BYTE 0: **conf_config**:
0xDF to enter boot-loader mode (DFU).
0xFF to reset configuration bytes to defaults.
0xB0 through 0xBF: Set ball type 0 through 15. Performs configuration reset after changing ball type.
0x01 to write configuration bytes.
Otherwise, device is reset after characteristic write with no changes. Ex. write 0x00 to cancel.

BYTE 1[7:4]: **conf_ball_type**: (value 0-15) (Default 0) Read only. Set this value using command 0xB0-0xBF (above).

BYTE 1[3:0]: **conf_data_packets**: (value 1-8) (Default 8)

BYTE 2: **conf_gyro_mag_on**: gyro magnitude data broadcasted (Value 0 or 1) (Default 1)

BYTE 3: **conf_accel_mag_on**: accelerometer magnitude data boradcasted (Value 0 or 1) (Default 0)

BYTE 4: **conf_auto_min**: Minimum sample number to start collision detection (Value 0 to 255) (Default 13 (0x0D) = 0.125 seconds)

BYTE 5: **conf_auto_threshold**: Threshold for magnitude of collision detection (Value 0 to 255) (Default 16 (0x10))

BYTE 6: **conf_compressed**: Compressed data (Value 0 or 1) (Default 1)

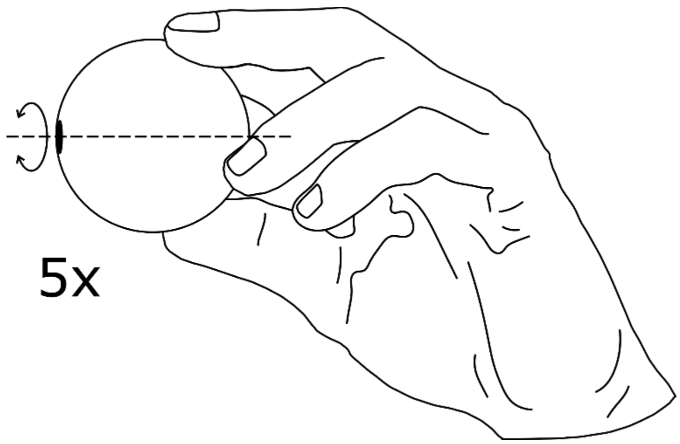
BYTE 7+: Done Care

Sample Configuration Byte String to write Defaults (Hex): 01 08 01 00 0D 10 01

Sample Configuration Byte String to change ball Type (Hex): B4

Configuration Gesture

Perform the following gesture to sync the DigiBall to your device: First, verify that the DigiBall is on and transmitting data. Next, hold the ball in such a way that allows you to partially rotate back and forth while keeping the black dot stationary. Hold the ball still for 3 seconds, and then rotate back and forth 5 times within 3 seconds. A BLE device with "DigiBall" will become available for connection.



Compliance Information

FCC ID: 2BP4A-170683C

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

(1) This device may not cause harmful interference, and
(2) This device must accept any interference received, including interference that may cause undesired operation.

UN38.3 Cleared for Air or Sea Transport, Battery and Sensor Assembly

Liability Disclaimer

Nathan Rhoades LLC reserves the right to make changes without further notice to the product to improve reliability, function or design. Nathan Rhoades LLC does not assume any liability arising out of the application or use of any product described herein.