



DigiCue Blue API

11/20/2024 Legacy (v1.1 and less)

Manufacturer Data ID: 0x03DE

Manufacturer Data: Advertisement (17 bytes)

Byte Index	Name [Bits]	Description
0	PACKET_COUNT	Packet Count 0-255: Increments after each shot, resets to 0 on button press or reset
1	CONFIG[7:5]	Reserved
	CONFIG[4:3]	Data Type 0 = Version 1 = Payload 2 = Tap Test 3 = Reserved
	CONFIG[2:1]	Mode (Sensitivity) 0 = Beginner 1 = Intermediate 2 = Advanced 3 = Custom
	CONFIG[0]	Buzz 0 = No Faults Occurred 1 = Faults Occurred
2	ACONF0[7]	State Freeze 0 = Off, 1 = On
	ACONF0[6]	State Shot Power 0 = Off, 1 = On
	ACONF0[5]	State Straightness 0 = Off, 1 = On
	ACONF0[4]	State Steering 0 = Off, 1 = On
	ACONF0[3]	State Follow Through Smoothness 0 = Off, 1 = On
	ACONF0[2]	State Jab 0 = Off, 1 = On
	ACONF0[1]	State Backstroke Pause 0 = Off, 1 = On
	ACONF0[0]	State Shot Pause 0 = Off, 1 = On
3	ACONF1[7:6]	Threshold Follow Through Smoothness 0 = 5 (Very Punchy) 1 = 7 (Punchy) 2 = 9 (Smooth) 3 = 11 (Very Smooth)
	ACONF1[5:4]	Threshold Jab 0 = 25 (Extreme) 1 = 50 (High) 2 = 75 (Medium) 3 = 100 (Low)
	ACONF1[3:2]	Threshold Backstroke Pause 0 = 8 (0.1s) 1 = 16 (0.2s) 2 = 41 (0.5s) 3 = 83 (1s)

	ACONF1[1:0]	Threshold Shot Interval 0 = 10 (5s) 1 = 16 (8s) 2 = 24 (12s) 3 = 30 (15s)
4	ACONF2[7:6]	Threshold Freeze 0 = 40 (1s) 1 = 80 (1.5s) 2 = 120 (2s) 3 = 160 (2.5s)
	ACONF2[5:4]	Threshold Shot Power 0 = 70 (Very Soft) 1 = 80 (Soft) 2 = 90 (Medium Soft) 3 = 100 (Medium)
	ACONF2[3:2]	Threshold Straightness 0 = 5 (Extreme) 1 = 10 (High) 2 = 20 (Medium) 3 = 35 (Low)
	ACONF2[1:0]	Threshold Steering 0 = 43 (Low) (15deg keepout) 1 = 32 (Medium) (30 deg keepout) 2 = 22 (High) (45 deg keepout) 3 = 11 (Extreme) (60 deg keepout)
5	ACONF3[0]	BUZZ_PASS_ENABLE = 0: Do Not Vibrate On Success (Default) BUZZ_PASS_ENABLE = 1: Vibrate Once Quickly On Success
	AFCON3[1]	BUZZ_ALARM_DISABLE = 0: Normal Operation (Default) BUZZ_ALARM_DISABLE = 1: Disable Vibrations For All Alarms
	ACONF3[2]	BUZZ_PATTERN_ENABLE = 0: Normal Vibration (Default) BUZZ_PATTERN_ENABLE = 1: Vibration Patterns On
	ACONF3[3]	BUZZ_PLAYBACK_ENABLE = 0: No playback on button press BUZZ_PLAYBACK_ENABLE = 1: Playback last fault based on vib. code
	ACONF3[7:4]	Reserved
6	ALERT0[7]	Alert Freeze 0 = No Fault, 1 = Fault
	ALERT0 [6]	Alert Shot Power 0 = No Fault, 1 = Fault
	ALERT0 [5]	Alert Straightness 0 = No Fault, 1 = Fault
	ALERT0 [4]	Alert Steering 0 = No Fault, 1 = Fault
	ALERT0 [3]	Alert Follow Through 0 = No Fault, 1 = Fault
	ALERT0 [2]	Alert Jab 0 = No Fault, 1 = Fault
	ALERT0 [1]	Alert Backstroke Pause 0 = No Fault, 1 = Fault
	ALERT0 [0]	Alert Shot Pause 0 = No Fault, 1 = Fault
7	ALERT1[7:1]	Freeze Magnitude (Cue Stillness)
	ALERT1[0]	Alert Steering Direction 0 = Left 1 = Right
8	SHOT_TIMER	Time since last shot 0-255: Increments of 0.512 seconds

9	PAUSE_TIME	Backstroke time 0-255: Increments of 12ms
10	FOLLOW_THRU	Follow through score 0-255: 0-12 is the typical range
11	JAB_MAG	Jab magnitude 0-255
12	IMPACT_ANGLE	Impact vector angle 0-255: 0-360 degrees in 1.40625 deg increments
13	IMPACT_MAG	Impact vector magnitude 0-255
14	FREEZE_ANG	Freeze vector angle 0-255: 0-360 degrees in 1.40625 deg increments
15	FREEZE_TIME	Freeze duration 0-160: Increments of 12ms
16	SHOT_POWER	Shot power 0-255: 60-110 is the typical range

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 0xBB 0x55
Full mac address	2:7	uint8	6	Full six-byte MAC address of DigiCue Blue (Legacy) device
RSSI	8	int8	1	Measured RSSI of DigiCue Blue (Legacy) device
Adv. Type	9	uint8	1	Always 0 = Advertisement from DigiCue Blue (Legacy)
Payload	10:26	uint8	17	Manufacturer data from DigiCue Blue (Legacy) device
Zeros	27:33	uint8	7	0
CRC16	34:35	uint16	2	CRC16 of bytes 0:25

Parsing Example (Python)

See <https://github.com/nataddrho/digicast> for more information

```
if len(mdata)==17: #DigiCue Blue manufacturer data is 17 bytes in length
```

```

#Check for reset
was_reset = False
if mdata[0]==0:
    was_reset = True
    for i in range(6,len(mdata)):
        if mdata[i]!=0:
            was_reset = False

config = mdata[1]
aconf0 = mdata[2]
aconf1 = mdata[3]
aconf2 = mdata[4]
aconf3 = mdata[5]
data_type = (config>>3)&3

if data_type<=1:

    alert0 = mdata[6]
    alert1 = mdata[7]
    shot_timer = mdata[8]
    pause_time = mdata[9]
    follow_thru = mdata[10]
    jab_mag = mdata[11]
    impact_angle = mdata[12] * 180 / 128.0
    impact_mag = mdata[13] / 255.0

```

```

freeze_ang = mdata[14] * 180 / 128.0
freeze_time = mdata[15]
freeze_mag = alert1 >> 1

score_finish = (freeze_time + 48) * 0.012
score_backstroke = pause_time * 0.012
tmp = jab_mag;
if (tmp > 125):
    tmp = 125
score_jab = (125 - tmp) / 12.5
score_follow = follow_thru - 1
if score_follow < 0:
    score_follow = 0
alert_steer_right = sin(impact_angle * pi / 180) < 0

steering_dir = alert1 & 1 == 1

tmp = impact_mag * 255
if tmp > 50:
    tmp = 50
score_straightness = (50 - tmp) / 5.0
score_steering = score_straightness * abs(cos(impact_angle * pi / 180))
score_interval = shot_timer / 255;

#Default thresholds (exceeding this value will generate a fault)
straightness_t = [0.893, 0.785, 0.571, 0.25]
steering_t = straightness_t
follow_t = [.4, .7, .9, 1]
jab_t = [.8, .6, .4, .2]
backstroke_t = [.1, .2, .5, 1]
interval_t = [0.0382, 0.0612, 0.0919, 0.114]
finish_t = [1/3, 1.5/3, 2/3, 2.5/3]

data = {}
data["Was Reset"] = was_reset
data["RSSI"] = rssi
data["MAC Address"] = mac_address
data["Straightness"] = score_straightness/10
data["Straightness Text"] = "%.1f"%score_straightness
data["Straightness Threshold"] = straightness_t[(aconf2>>2)&3]
data["Straightness Enabled"] = (aconf0>>5)&1==1
data["Straightness Angle"] = impact_angle
data["Finish"] = score_finish/3
data["Finish Text"] = "%.1fs"%(score_finish)
data["Finish Threshold"] = finish_t[(aconf2>>6)&3]
data["Finish Enabled"] = (aconf0>>7)&1==1
data["Tip Steer"] = score_steering/10
data["Tip Steer Text"] = "%.1f"%score_steering
data["Tip Steer Threshold"] = None
data["Tip Steer Enabled"] = (aconf0>>4)&1==1
data["Follow Through"] = score_follow/10
data["Follow Through Text"] = "%.1f"%score_follow
data["Follow Through Threshold"] = follow_t[(aconf1>>6)&3]
data["Follow Through Enabled"] = (aconf0>>3)&1==1
data["Jab"] = score_jab/10
data["Jab Text"] = "%.1f"%score_jab
data["Jab Threshold"] = jab_t[(aconf1>>4)&3]
data["Jab Enabled"] = (aconf0>>2)&1==1
data["Backstroke Pause"] = score_backstroke
data["Backstroke Pause Text"] = "%.1fs"%score_backstroke
data["Backstroke Pause Threshold"] = backstroke_t[(aconf1>>2)&3]
data["Backstroke Pause Enabled"] = (aconf0>>1)&1==1
data["Shot Interval"] = score_interval
data["Shot Interval Text"] = "%is"%(shot_timer * 0.512)
data["Shot Interval Threshold"] = interval_t[aconf1&3]
data["Shot Interval Enabled"] = aconf0&1==1

```



Compliance Information

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.

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.