



14

[ホーム](#)[タイムライン](#)[トレンド](#)[質問](#)[公式イベント](#)[募集](#)[Organization](#)[Qiita Blog](#)

記事投稿した全員がZoomステッカーが貰える！ZoomAPI/SDKを使ってZoomを拡張してみよう！



⚠ この記事は最終更新日から3年以上が経過しています。



@mozzio369

投稿日 2018年06月29日 更新日 2018年07月02日

# トイドローン Tello でプログラミング！バイナリーコマンドの使い方



Python, ドローン, tello



Ryze Technologyのトイドローン Tello をプログラミングで操作するためのバイナリーコマンドの使い方です。Pythonで説明してますが、ソケット通信やバイト操作ができれば何でもOKです。

## バイナリーコマンドの基本

Telloとの通信にはUDPを使います。ソケットを作成し、コマンドを送信します。

```
import socket
```

```
TELLO_IP = '192.168.10.1'
```

```
TELLO_PORT_CMD = 8889
```

```
TELLO_PORT_VIDEO = 6038
```

```
addr = (TELLO_IP, TELLO_PORT_CMD)
```

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

```
sock.sendto(cmd, addr)
```

公式のTello SDKでは一部のコマンド、一部の機能しか公開されていませんが、DJI Ryze Tello Drone Forumなど世界のTello Pilotsの努力でバイナリーコマンドの仕様がほぼ明らかになっています。非公式情報なので、あくまで自己責任でお試しを。

最初にTelloと接続するためのコマンドは以下のように作成します。ポート番号は、ビデオストリーム受信のためのポート番号で、デフォルトでは6038です。2バイトのリトルエンディアンで設定します。

```
TELLO_PORT = 6038
cmd = 'conn_req:'.encode() + TELLO_PORT_VIDEO.to_bytes(2, 'little')
```

他のコマンドは、バイト列からStructを使ってパック化します。以下は、テイクオフコマンドの例です。

```
from struct import Struct
cmd_takeoff = [0xcc, 0x58, 0x00, 0x7c, 0x68, 0x54, 0x00, 0x01, 0x00, 0x6a, 0x90]
s = Struct("!11B")
cmd = s.pack(*cmd_takeoff)
```

## バイナリーコマンドの作り方

### フォーマット

Telloのコマンドフォーマットです。

0	1 ~ 2	3	4	5 ~ 6	7 ~ 8	---	N-1 ~ N
先頭バイト	パケット長	CRC8	パケットタイプ	コマンドID	シーケンス番号	データ	CRC16

### 先頭バイト

先頭バイトは 0xCC 固定です。

## パケット長

パケット長は、バイト1~2の先頭13ビットで、下位3ビットは未使用(0)になります。テイクオフコマンドの場合は、データ部分がないのでパケット長は11バイトとなり、以下のようになります。

```
>>> hex(11<<3)
'0x58'
```

これをバイト1~2にリトルエンディアンで 0x58 0x00 と設定します。

以降、複数バイトからなるフィールドは、すべてリトルエンディアンです。

## CRC8

バイト3は、先頭バイトからパケット長までの3バイトのCRC8です。ここでは先人の知恵をそのまま拝借して、テーブルを使って計算する方法を紹介します。

```
TBL_CRC8 = [
    0x00, 0x5e, 0xbc, 0xe2, 0x61, 0x3f, 0xdd, 0x83, 0xc2, 0x9c, 0x7e, 0x20, 0xa3, 0xfd,
    0x9d, 0xc3, 0x21, 0x7f, 0xfc, 0xa2, 0x40, 0x1e, 0x5f, 0x01, 0xe3, 0xbd, 0x3e, 0x60,
    0x23, 0x7d, 0x9f, 0xc1, 0x42, 0x1c, 0xfe, 0xa0, 0xe1, 0xbf, 0x5d, 0x03, 0x80, 0xde,
    0xbe, 0xe0, 0x02, 0x5c, 0xdf, 0x81, 0x63, 0x3d, 0x7c, 0x22, 0xc0, 0x9e, 0x1d, 0x43,
    0x46, 0x18, 0xfa, 0xa4, 0x27, 0x79, 0x9b, 0xc5, 0x84, 0xda, 0x38, 0x66, 0xe5, 0xbb,
    0xdb, 0x85, 0x67, 0x39, 0xba, 0xe4, 0x06, 0x58, 0x19, 0x47, 0xa5, 0xfb, 0x78, 0x26,
    0x65, 0x3b, 0xd9, 0x87, 0x04, 0x5a, 0xb8, 0xe6, 0xa7, 0xf9, 0x1b, 0x45, 0xc6, 0x98,
    0xf8, 0xa6, 0x44, 0x1a, 0x99, 0xc7, 0x25, 0x7b, 0x3a, 0x64, 0x86, 0xd8, 0x5b, 0x05,
    0x8c, 0xd2, 0x30, 0x6e, 0xed, 0xb3, 0x51, 0x0f, 0x4e, 0x10, 0xf2, 0xac, 0x2f, 0x71,
    0x11, 0x4f, 0xad, 0xf3, 0x70, 0x2e, 0xcc, 0x92, 0xd3, 0x8d, 0x6f, 0x31, 0xb2, 0xec,
    0xaf, 0xf1, 0x13, 0x4d, 0xce, 0x90, 0x72, 0x2c, 0x6d, 0x33, 0xd1, 0x8f, 0x0c, 0x52,
    0x32, 0x6c, 0x8e, 0xd0, 0x53, 0x0d, 0xef, 0xb1, 0xf0, 0xae, 0x4c, 0x12, 0x91, 0xcf,
    0xca, 0x94, 0x76, 0x28, 0xab, 0xf5, 0x17, 0x49, 0x08, 0x56, 0xb4, 0xea, 0x69, 0x37,
    0x57, 0x09, 0xeb, 0xb5, 0x36, 0x68, 0x8a, 0xd4, 0x95, 0xcb, 0x29, 0x77, 0xf4, 0xaa,
    0xe9, 0xb7, 0x55, 0x0b, 0x88, 0xd6, 0x34, 0x6a, 0x2b, 0x75, 0x97, 0xc9, 0x4a, 0x14,

    0x71, 0x23, 0xc8, 0x96, 0x15, 0x4b, 0x29, 0xf7, 0xb6, 0x68, 0x22, 0x54, 0xd7, 0x89,
```

```
0x74, 0x28, 0x00, 0x00, 0x10, 0x40, 0x80, 0x17, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
]
```

```
def calc_crc8(buf, size):
    i = 0
    seed = 0x77
    while size > 0:
        seed = TBL_CRC8[(seed ^ buf[i]) & 0xff]
        i = i + 1
        size = size - 1
    return seed

buf = bytearray()
for b in cmd[:3]:
    buf.append(b)

crc8 = calc_crc8(buf, len(buf))
```

## パケットタイプ・コマンドID

バイト4はパケットタイプです。バイト5~6はコマンドIDです。離着陸およびジョイスティックに関するコマンドは以下です。

Command	Packet Type	Command ID
Takeoff	0x68 (104)	0x54 (84)
Land	0x68 (104)	0x55 (85)
Stick (Timestamp)	0x60 (96)	0x50 (80)

他にもたくさんのコマンドがあります。興味ある方は、以下のサイトがよくまとまっています。



## シーケンス番号

バイト7~8はシーケンス番号です。基本的にインクリメントですが、Stick (Timestamp) コマンドや、H.264ビデオストリームのデコードに必要なSPS/PPSリクエストコマンドなど、一部のコマンドは常時0になります。

## データ

Takeoffコマンドでは、データ部分はありません。

Landコマンドではデータは 0x00 の1バイトです。

Stick (Timestamp)では6バイトのジョイスティック情報と5バイトのタイムスタンプ情報が付きます。

ジョイスティック情報の6バイト=48ビットのうち、最初の4ビットは fast (1) か normal(0) かのモード設定です。残りの44ビットがジョイスティックの操作に割り当てられています。

b1 ~ b4	b5 ~ b15	b16 ~ b26	b27 ~ b37	b38 ~ b48
mode	yaw (回転)	throttle (上下)	pitch (前後)	role (左右)

ジョイスティックの操作は、何もしない場合が1024で、±660の範囲で設定されます。例えば normalモードで、ジョイスティックを何も操作しないホバリング状態の場合は以下のようになります。

b1 ~ b4	b5 ~ b15	b16 ~ b26	b27 ~ b37	b38 ~ b48
0000	100000000000	100000000000	100000000000	100000000000

バイト単位で表すと 0x08 0x01 0x00 0x20 0x04 0x00 となり、これをバイト9~14 にリトルエンディアンで設定します。

タイムスタンプ情報は、時（1バイト）、分（1バイト）、秒（1バイト）、ミリ秒（2バイト）の計5バイトです。

これらをまとめると、Stick (Timestamp)コマンドは以下のように作成できます（最後のCRC16を除く）。

```
import datetime

cmd = [0xcc, 0xb0, 0x00, 0x7f, 0x60, 0x50, 0x00, 0x00, 0x00] # 先頭バイトからシーケンス
mode = 0

yaw = 1024
thr = 1024
pitch = 1024
roll = 1024

stick_data = (mode<<44) + (yaw<<33) + (thr<<22) + (pitch<<11) + roll
for i in range(0,6):
    cmd.append(stick_data>>(8*i) & 0xff)

now = datetime.datetime.now()

h = now.hour
m = now.minute
s = now.second
ms = round(now.microsecond / 1000)

cmd.append(h)
cmd.append(m)
cmd.append(s)
cmd.append(ms & 0xff)
cmd.append(ms >> 8)
```

## CRC16

最後の2バイトはCRC16です。こちらもCRC8と同様、先人の知恵をそのまま拝借します。下記で計算したCRC16を最後の2バイトに追加して、コマンドの完成です。

```
# CTC16 Table
TBL_CRC16 = [
    0x0000, 0x1189, 0x2312, 0x329b, 0x4624, 0x57ad, 0x6536, 0x74bf, 0x8c48, 0x9dc1, 0xaf5a, 0xbfe3, 0xc54c, 0xd4e5,
    0x1081, 0x0108, 0x3393, 0x221a, 0x56a5, 0x472c, 0x75b7, 0x643e, 0x9cc9, 0x8d40, 0xbfb3, 0xae58, 0x95e3, 0x846a,
    0x2102, 0x308b, 0x0210, 0x1399, 0x6726, 0x76af, 0x4434, 0x55bd, 0xad4a, 0xbcc3, 0x8e58, 0xff51, 0xc6dc, 0xd755,
    0x3183, 0x200a, 0x1291, 0x0318, 0x77a7, 0x662e, 0x54b5, 0x453c, 0xbdc3, 0xac42, 0x9e51, 0x8f58, 0xf5e3, 0xe46a,
    0x4204, 0x538d, 0x6116, 0x709f, 0x0420, 0x15a9, 0x2732, 0x36bb, 0xce4c, 0xdfc5, 0xed58, 0xfcd3, 0x86dc, 0x9755,
    0x5285, 0x430c, 0x7197, 0x601e, 0x14a1, 0x0528, 0x37b3, 0x263a, 0xdec3, 0xcfc4, 0xf551, 0xe458, 0x9663, 0x876a,
    0x6306, 0x728f, 0x4014, 0x519d, 0x2522, 0x34ab, 0x0630, 0x17b9, 0xef4e, 0xfec7, 0xc651, 0xd758, 0x8563, 0x946a,
    0x7387, 0x620e, 0x5095, 0x411c, 0x35a3, 0x242a, 0x16b1, 0x0738, 0xffcf, 0xee46, 0xdc51, 0xcd58, 0x9f63, 0x8e6a,
    0xa808, 0x9581, 0x8712, 0x769b, 0x4024, 0x51ad, 0x6336, 0x72bf, 0x0c48, 0x1dc1, 0x2f5a, 0x3fe3, 0x454c, 0x54e5,
    0x9101, 0x8008, 0x7293, 0x631a, 0x37a5, 0x262c, 0x14b7, 0x053e, 0xb548, 0xa4c1, 0x965a, 0x8759, 0xf5e3, 0xe46a,
    0xc673, 0xd77a, 0x8101, 0x7008, 0x6293, 0x531a, 0x27a5, 0x162c, 0x04b7, 0x153e, 0x4548, 0x54c1, 0x665a, 0x7759, 0x4963, 0x586a,
    0x6af3, 0x7b7a, 0x0801, 0x1908, 0x2b93, 0x3a1a, 0x4ea5, 0x5f2c, 0x6db7, 0x7c3e, 0x8c48, 0x9dc1, 0xaf5a, 0xbfe3, 0xc54c, 0xd4e5,
    0x1081, 0x0108, 0x3393, 0x221a, 0x56a5, 0x472c, 0x75b7, 0x643e, 0x9cc9, 0x8d40, 0xbfb3, 0xae58, 0x95e3, 0x846a, 0x7273, 0x637a,
    0x2102, 0x308b, 0x0210, 0x1399, 0x6726, 0x76af, 0x4434, 0x55bd, 0xad4a, 0xbcc3, 0x8e58, 0xff51, 0xc6dc, 0xd755, 0xed58, 0xfcd3,
    0x3183, 0x200a, 0x1291, 0x0318, 0x77a7, 0x662e, 0x54b5, 0x453c, 0xbdc3, 0xac42, 0x9e51, 0x8f58, 0xf5e3, 0xe46a, 0x9663, 0x876a,
    0x4204, 0x538d, 0x6116, 0x709f, 0x0420, 0x15a9, 0x2732, 0x36bb, 0xce4c, 0xdfc5, 0xed58, 0xfcd3, 0x86dc, 0x9755, 0x6563, 0x546a,
    0x5285, 0x430c, 0x7197, 0x601e, 0x14a1, 0x0528, 0x37b3, 0x263a, 0xdec3, 0xcfc4, 0xf551, 0xe458, 0x9663, 0x876a, 0x5573, 0x447a,
    0x6306, 0x728f, 0x4014, 0x519d, 0x2522, 0x34ab, 0x0630, 0x17b9, 0xef4e, 0xfec7, 0xc651, 0xd758, 0x8563, 0x946a, 0x6273, 0x537a,
    0x7387, 0x620e, 0x5095, 0x411c, 0x35a3, 0x242a, 0x16b1, 0x0738, 0xffcf, 0xee46, 0xdc51, 0xcd58, 0x9f63, 0x8e6a, 0x5c73, 0x4d7a,
    0xa808, 0x9581, 0x8712, 0x769b, 0x4024, 0x51ad, 0x6336, 0x72bf, 0x0c48, 0x1dc1, 0x2f5a, 0x3fe3, 0x454c, 0x54e5, 0x8273, 0x737a,
    0x9101, 0x8008, 0x7293, 0x631a, 0x37a5, 0x262c, 0x14b7, 0x053e, 0xb548, 0xa4c1, 0x965a, 0x8759, 0xf5e3, 0xe46a, 0x9663, 0x876a,
    0x6563, 0x546a, 0x5285, 0x430c, 0x7197, 0x601e, 0x14a1, 0x0528, 0x37b3, 0x263a, 0xdec3, 0xcfc4, 0xf551, 0xe458, 0x9663, 0x876a, 0x5573, 0x447a,
    0x6306, 0x728f, 0x4014, 0x519d, 0x2522, 0x34ab, 0x0630, 0x17b9, 0xef4e, 0xfec7, 0xc651, 0xd758, 0x8563, 0x946a, 0x6273, 0x537a,
    0x7387, 0x620e, 0x5095, 0x411c, 0x35a3, 0x242a, 0x16b1, 0x0738, 0xffcf, 0xee46, 0xdc51, 0xcd58, 0x9f63, 0x8e6a, 0x5c73, 0x4d7a,
    0xa808, 0x9581, 0x8712, 0x769b, 0x4024, 0x51ad, 0x6336, 0x72bf, 0x0c48, 0x1dc1, 0x2f5a, 0x3fe3, 0x454c, 0x54e5, 0x8273, 0x737a,
    0x9101, 0x8008, 0x7293, 0x631a, 0x37a5, 0x262c, 0x14b7, 0x053e, 0xb548, 0xa4c1, 0x965a, 0x8759, 0xf5e3, 0xe46a, 0x9663, 0x876a, 0x6563, 0x546a,
    0x5285, 0x430c, 0x7197, 0x601e, 0x14a1, 0x0528, 0x37b3, 0x263a, 0xdec3, 0xcfc4, 0xf551, 0xe458, 0x9663, 0x876a, 0x5573, 0x447a,
    0x6306, 0x728f, 0x4014, 0x519d, 0x2522, 0x34ab, 0x0630, 0x17b9, 0xef4e, 0xfec7, 0xc651, 0xd758, 0x8563, 0x946a, 0x6273, 0x537a,
    0x7387, 0x620e, 0x5095, 0x411c, 0x35a3, 0x242a, 0x16b1, 0x0738, 0xffcf, 0xee46, 0xdc51, 0xcd58, 0x9f63, 0x8e6a, 0x5c73, 0x4d7a,
    0xa808, 0x9581, 0x8712, 0x769b, 0x4024, 0x51ad, 0x6336, 0x72bf, 0x0c48, 0x1dc1, 0x2f5a, 0x3fe3, 0x454c, 0x54e5, 0x8273, 0x737a,
    0x9101, 0x8008, 0x7293, 0x631a, 0x37a5, 0x262c, 0x14b7, 0x053e, 0xb548, 0xa4c1, 0x965a, 0x8759, 0xf5e3, 0xe46a, 0x9663, 0x876a, 0x6563, 0x546a,
    0x5285, 0x430c, 0x7197, 0x601e, 0x14a1, 0x0528, 0x37b3, 0x263a, 0xdec3, 0xcfc4, 0xf551, 0xe458, 0x9663, 0x876a, 0x5573, 0x447a,
    0x6306, 0x728f, 0x4014, 0x519d, 0x2522, 0x34ab, 0x0630, 0x17b9, 0xef4e, 0xfec7, 0xc651, 0xd758, 0x8563, 0x946a, 0x6273, 0x537a,
    0x7387, 0x620e, 0x5095, 0x411c, 0x35a3, 0x242a, 0x16b1, 0x0738, 0xffcf, 0xee46, 0xdc51, 0xcd58, 0x9f63, 0x8e6a, 0x5c73, 0x4d7a,
    0xa808,
```

```
0x0400, 0x0501, 0x071a, 0x0000, 0x022c, 0x0500, 0xe10e, 0x1007, 0x0040, 0x10c0, 0x21
0x9489, 0x8500, 0xb79b, 0xa612, 0xd2ad, 0xc324, 0xf1bf, 0xe036, 0x18c1, 0x0948, 0x31
0xa50a, 0xb483, 0x8618, 0x9791, 0xe32e, 0xf2a7, 0xc03c, 0xd1b5, 0x2942, 0x38cb, 0x0i
0xb58b, 0xa402, 0x9699, 0x8710, 0xf3af, 0xe226, 0xd0bd, 0xc134, 0x39c3, 0x284a, 0x1i
0xc60c, 0xd785, 0xe51e, 0xf497, 0x8028, 0x91a1, 0xa33a, 0xb2b3, 0x4a44, 0x5bcd, 0x6c
0xd68d, 0xc704, 0xf59f, 0xe416, 0x90a9, 0x8120, 0xb3bb, 0xa232, 0x5ac5, 0x4b4c, 0x7f
0xe70e, 0xf687, 0xc41c, 0xd595, 0xa12a, 0xb0a3, 0x8238, 0x93b1, 0x6b46, 0x7acf, 0x4f
0xf78f, 0xe606, 0xd49d, 0xc514, 0xb1ab, 0xa022, 0x92b9, 0x8330, 0x7bc7, 0x6a4e, 0x5f
]
```

```
def calc_crc16(buf, size):
    i = 0
    seed = 0x3692
    while size > 0:
        seed = TBL_CRC16[(seed ^ buf[i]) & 0xff] ^ (seed >> 8)
        i = i + 1
        size = size - 1
    return seed

buf = bytearray()
for b in cmd[:len(cmd)-2]:
    buf.append(b)

crc16 = calc_crc16(buf, len(buf))
```

## サンプル

TelloをMacのキーボードで操作し、OpenCVによるカメラ映像のキャプチャ、顔認識+自動追従などができるプログラムを以下の記事で紹介しています。

- [トイドローン Tello をMacで操作&カメラ映像をOpenCVで受信してみる](#)
- [トイドローン Tello をプログラミングで機能拡張！顔認識と自動追尾を実装してみた](#)

## 参考

Telloユーザのコミュニティです。おもしろいです。