

# FruitPunch スクールセットガイド (基本編)

## 目次

1. 内容.....	2
2. 準備物 .....	2
3. 機材のセットアップ .....	2
3-1. スクールセットのセットアップ .....	2
4. FruitPunch と Tello の接続 .....	2
4-1. Tello の準備 .....	2
4-2. FruitPunch と Tello の接続.....	3
4-3. FruitPunch と Tello の再接続.....	4
5. プログラミング .....	4
5-1. プログラムで Tello を制御する仕組み .....	4
5-2. BASIC コマンドを実行する .....	5
5-3. BASIC コマンドを使って FruitPunch にメッセージを送る .....	5
5-4. プログラムの作成と実行 .....	6
5-5. プログラムの修正 .....	7
5-6. プログラムの保存と読出 .....	8
5-7. プログラムの消去 .....	8
5-8. プログラムの自動実行.....	8
6. ショートカットキー .....	9

# 1. 内容

本ガイドは、IchigoIgai スクールセットを使って Tello・Tello EDU の BASIC 言語プログラミングを行うためのセットアップ方法や、IchigoDake の基本的な使い方・プログラミング方法について、解説しています。初めて IchigoIgai スクールセットでの Tello・Tello EDU プログラミングを行う方は、まず本ガイドをご一読ください。

## 2. 準備物

- Tello EDU ×1
- FruitPunch ×1
- DakeJacket ×1
- IchigoDake (BASIC 版) ×1
- IchigoIgai スクールセット ×1
- モバイルバッテリー ×1

## 3. 機材のセットアップ

Tello のプログラミングを行うための各機材を準備します。

### 3-1. スクールセットのセットアップ

「FruitPunch 接続説明書(スクールセット編)」(※1)に従って、スクールセットのセットアップを行います。

(※1) <https://na-s.jp/FruitPunch/docs/fruitpunch-howto1-ss.pdf>

## 4. FruitPunch と Tello の接続

Tello は Wi-Fi のアクセスポイントとして動作するドローンです。FruitPunch を使って Tello のアクセスポイントに接続すれば、IchigoDake のプログラムで Tello を制御できるようになります。

ここでは、FruitPunch と Tello を接続する方法について解説します。

### 4-1. Tello の準備

初めて利用する Tello, Tello EDU の機体には、アクティベーションが必要です。アクティベーションには iOS や Android の Tello アプリを使用してください。

アクティベーションが完了したら、Tello のアクセスポイント名(SSID)を確認します。SSID は Tello のバッテリーを差し込む場所に記載されています。

FruitPunch から Tello に接続するためには、この SSID が必要となるので、メモしておきます。

Tello にバッテリーを差し込み、電源を入れてカメラ右上の LED(機体ステータスインジケーター)が黄色点滅になるのを待ちます。(※1)

黄色点滅になったら、FruitPunch から Tello に接続できます。

(※1) 黄色点滅にならない場合は Tello のユーザーマニュアルをご参照ください。

## 4-2. FruitPunch と Tello の接続

まずは FruitPunch が接続可能なアクセスポイントを調べます。

Tello の電源が入っている状態で、キーボードで以下のコマンドを打ち、エンターキーを押してください。

```
? "FP APL"
```

少し待つと、以下のように、FruitPunch から接続可能な SSID の一覧が画面に表示されます。

```
' TELLO-XXXXXX  
' 10ASEF_AAAAA  
' OK
```

一覧の中に、接続したい Tello の SSID があることを確認してください(※1)。

次に、FruitPunch と Tello を接続します。

キーボードで以下のコマンドを打ち、エンターキーを押してください。

```
? "FP APL SSID"
```

「SSID」の部分には、接続したい Tello の SSID を差し込んでください。例えば、SSID が「TELLO-012345」ならば以下ようになります。

```
? "FP APL TELLO-012345"
```

エンターキーを押してからしばらく待ち、画面に以下のような表示が出ると、接続成功です。FruitPunch の緑 LED が点灯します。

```
' . . . . .  
' WiFi connected 192.168.10.2  
' OK
```

接続に失敗した場合は、以下のような表示となります。FruitPunch の緑 LED は点滅します。

```
' . . . . .  
' WiFi error
```

接続に失敗した場合は、Tello の電源が入っているか、SSID に間違いはないか、Tello のバッテリーは充分かなどを確認の上、再度、本手順をくり返してください。

接続に成功した後は、Tello を制御するプログラムで、Tello を飛行させることができるようになります。

(※1)ときどき本来の SSID よりも僅かに一部の文字が多かったり、少なかったりすることがあります。しかし実際の動作には影響ありません。

### 4-3. FruitPunch と Tello の再接続

FruitPunch は一度アクセスポイントに接続すると、自動的にその SSID を記憶します。

FruitPunch は、起動した際に前回接続した SSID があるとき、自動的にその SSID へ接続するようになっています。

そのため、「FruitPunch と Tello の接続」の方法で Tello と接続した後は、

1. Tello の電源を入れる
2. FruitPunch のスイッチを入れ、起動する

という手順のみで、前回接続した Tello に接続できます。

## 5. プログラミング

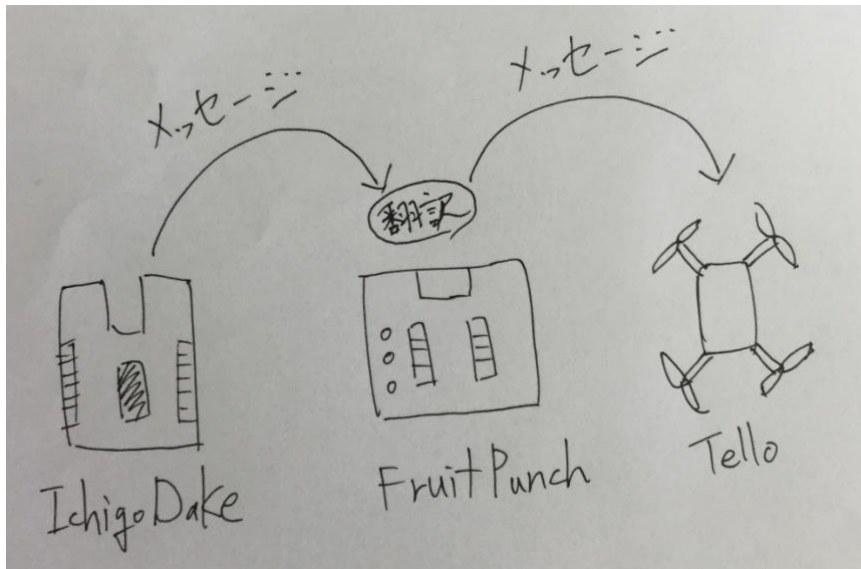
FruitPunch と Tello が接続されていると、IchigoDake で実行したコマンドやプログラムに応じて、Tello を様々に制御できます。

ここでは、IchigoDake で各種の BASIC コマンドを実行する方法や、プログラムを作成・実行する方法について解説します。

### 5-1. プログラムで Tello を制御する仕組み

Tello は接続されている機器から制御用のメッセージ(Tello コマンド)を受け取ると、その通りに飛行するという機能を持っています。

IchigoDake が FruitPunch にメッセージを送ると、FruitPunch はそれを適切に翻訳し、Tello に送ります。FruitPunch へのメッセージはプログラムによって送られます。つまりプログラムを利用してメッセージの「内容」と「送るタイミング」を制御することで、Tello を様々に動かせるようになります。



## 5-2. BASIC コマンドを実行する

IchigoDake はキーボードを使って操作するコンピュータです。キーボードで BASIC コマンドを打ち、エンターキーで BASIC コマンドを実行する、という形で使用します。

例えば、IchigoDake の表面にある LED を光らせることができる「LED1」という BASIC コマンドがあります。

この BASIC コマンドを実行するには、まずキーボードで以下のように BASIC コマンドを打ちます。

```
LED1
```

BASIC コマンドを打ち終わったら、エンターキーを押します。エンターキーを押すことで、カーソルがいる行に書かれている BASIC コマンドを実行できます。

IchigoDake(BASIC 版)で利用できる BASIC コマンドは「IchigoJam BASIC リファレンス」(※1)をご覧ください。

(※1) <https://ichigojam.net/IchigoJam.html>

## 5-3. BASIC コマンドを使って FruitPunch にメッセージを送る

?コマンドを使うと、IchigoDake から FruitPunch にメッセージを送ることができます。

```
? "MESSAGE"
```

「MESSAGE」の部分で、送りたいメッセージに差し替えて使います。

例えば、以下のような文字を打ち、エンターキーを押して?コマンドを実行します。

```
? "HELLO"
```

すると、モニタに「HELLO」という文字が新しく表示されます。?コマンドには「FruitPunch にメッセージを送る」と同時に「画面に文字を出す」という機能もあるからです。

しかし、Tello には何も起きません。なぜなら FruitPunch は「HELLO」というメッセージの意味が理解できないからです。FruitPunch は意味の分からないメッセージを受け取っても、何もしてくれません。

Tello を制御するためには、ちゃんと FruitPunch が理解できるようなメッセージを送らなくてはなりません。つまり、送るメッセージの内容は、FruitPunch のコマンドになっていなくてはならないのです。

FruitPunch のコマンドを送ると、FruitPunch は Tello に向けてメッセージを送ってくれます。

例えば FruitPunch のコマンドのひとつに「FP QRUN」というものがあります。

このコマンドを FruitPunch に送ると、Tello が離陸し、しばらくすると着陸します。

Tello が飛ばない場合は下記をお試しください。

- ・ Tello の電源が切れていないか確認してください。
- ・ Tello を再起動して、Tello のカメラ右上の LED が黄色点滅になってから 10 秒ほど後に再度お試しください。

```
? "FP QRUN"
```

FruitPunch のコマンドは色々なものがあります。

どのようなコマンドがあるかは、「リファレンスマニュアル」(※1)をご覧ください。

(※1)[https://na-s.jp/FruitPunch/docs/manual\\_1.1.0\\_ja.html](https://na-s.jp/FruitPunch/docs/manual_1.1.0_ja.html)

#### 5-4. プログラムの作成と実行

プログラムとは、どんな BASIC コマンドをどんな順番で実行するかを記述したものです。

プログラムを作れば、たくさんのコマンドを決まった順番でまとめて実行できるようになります。また、プログラムを一度作ってしまえば、何度でも実行できるようになります。

プログラムを作るには、実行したい BASIC コマンドの最初に数字を入れて、エンターキーを押します。この数字のことを行番号と呼びます。

例えば、以下のように文字を打ち、エンターキーを押します。

```
10 ? "FP QRUN"
```

これでプログラムが作られます。

しかしプログラムは作られただけでは何も起きません。プログラムを実行するには、「RUN」というコマンドを使います。

```
RUN
```

もう少しプログラムの中身を増やしてみましょう。例えば、以下のように文字を打ってから、エンターキーを押します。

```
5 ? "FP QF 50"
```

これでプログラムに上記のコマンドが追加されます。

現在、プログラムがどんな形をしているかは「LIST」コマンドで確認できます。

## LIST

すると、画面には以下のような文字が新しく表示されます。これが、現在作られているプログラムの内容です。

```
5 ?"FP QF 50"  
10 ?"FP QRUN"
```

行番号は、プログラムの中でコマンドが実行される順番を表しています。数字が小さいものから順番にコマンドが実行されるわけです。このプログラムでは、まず「?"FP QF 50"」が実行され、続けて「?"FP QRUN"」が実行される、ということになります。

行番号の値は、1 ずつ増やしていく必要はありません。一般的には、最初を 10 として、それから 10 ずつ増やしていくことが推奨されます。これは、後でプログラムを修正したくなったとき、行と行の間に新しい行を入れられるようにしておくためです。

「RENUM」コマンドを使うと、現在作っているプログラムの行番号が自動的に整理されます。

## RENUM

RENUM コマンドを実行したあとで LIST コマンドを実行すると、行番号が変化していることが分かります。しかし順番自体は変化していません。

```
10 ?"FP QF 50"  
20 ?"FP QRUN"
```

実際に、RUN コマンドでプログラムを実行してみましょう。ここで作ったプログラムでは、Tello は離陸したあと、前方に約 50cm 進み、最後に着陸します。

### 5-5. プログラムの修正

作ったプログラムを後から修正することもできます。

例えば、行番号 10 にある BASIC コマンドを?"FP QF 100"に修正したい場合は、以下のように文字を打ち、エンターキーを押します。

```
10 ?"FP QF 100"
```

エンターキーを押さなければ修正されないことに注意してください。

ちゃんと修正されたかどうかは、LIST コマンドで確認できます。

プログラムの特定の部分を削除したい場合は、削除したい行番号だけを打ち、エンターキーを押します。

```
10
```

これで行番号が 10 の部分が削除されます。削除されたかどうかは、やはり LIST コマンドで確認できます。

## 5-6. プログラムの保存と読出

一度作ったプログラムは RUN コマンドを使って何度でも実行できます。しかし FruitPunch や DakeJacket の電源を切ったり IchigoDake を DakeJacket から引き抜いたりすると、プログラムは消えてしまいます。

「SAVE」コマンドを使えば作ったプログラムを保存しておくことができます。

### SAVE 0

IchigoDake には 4 つの保存先が用意されており、「SAVE」の後に書く数字(0 ~ 3 のいずれかを)によって、保存先を変えられます。

SAVE コマンドでプログラムを保存しておけば、電源を切った後でも、いつでもプログラムを読み出して、実行できる状態に戻せます。

プログラムを読み出すには「LOAD」コマンドを使います。

### LOAD 0

「LOAD」の後に書く数字で、読み出すプログラムの保存先(0 ~ 3 のいずれかを)を指定します。

LOAD コマンドを実行した後で、LIST コマンドを実行すれば、プログラムが読み出されたことを確認できますし、RUN コマンドで実行することもできます。

読み出した後のプログラムの修正もできます。ただし、修正した後のプログラムは、再度 SAVE コマンドを使わないと保存されないことに注意してください。

## 5-7. プログラムの消去

作ったプログラムは「NEW」コマンドで消去できます。一度プログラムを作った後、最初から別のプログラムを作りたいというときに利用します。

### NEW

NEW コマンドでは、0 ~ 3 番の保存先にあるプログラムは消えません。

## 5-8. プログラムの自動実行

SAVE コマンドを使って 0 番に保存しておいたプログラムは、キーボードの操作なしに実行することが可能です。これを自動実行といいます。

プログラムを自動実行するには、IchigoDake にある白いボタンを押したまま、DakeJacket に差し込みます。

このとき、FruitPunch と DakeJacket のスイッチは ON にしておきます。

自動実行の機能を使って実行するようなプログラムは、最初に空のメッセージを送るようにしておくといいでしょう。なぜなら、自動実行の場合ははじめに空のメッセージを送らないと、FruitPunch へのメッセージがうまく送られない可能性があるためです。

例えば、以下の 1 行を追加します。

### 5 ?" "



## 6. ショートカットキー

幾つかの BASIC コマンドには、ショートカットキーが用意されています。ショートカットキーを使うとわざわざ文字を打たなくても、すぐにコマンドを実行できます。

ここでは代表的なショートカットキーを紹介します。

- F1 キー

「CLS」コマンドのショートカットキーです。CLS コマンドは画面に表示されている文字を全て消去する BASIC コマンドです。あくまで画面上の文字を消すだけなので、作成中のプログラムは消去されません。F1 キーを押すと、CLS コマンドが即座に実行されます。

- F2 キー

LOAD コマンドのショートカットキーです。

F2 キーを押すと、画面に「LOAD」の文字が現れます。しかし実行はされません。

実行するには、F2 キーの後で数字を打ち、エンターキーを押します。

- F3 キー

SAVE コマンドのショートカットキーです。

F3 キーを押すと、画面に「SAVE」の文字が現れます。しかし実行はされません。

実行するには、F3 キーの後で数字を打ち、エンターキーを押します。

- F4 キー

LIST コマンドのショートカットキーです。

F4 キーを押すと即座に LIST コマンドが実行されます。

- F5 キー

RUN コマンドのショートカットキーです。

F5 キーを押すと即座に RUN コマンドが実行されます。

### 代表的なショートカットキー一覧

ショートカットキー	対応するコマンド	コマンドの意味	即時実行
F1	CLS	画面の文字を消去	○
F2	LOAD	プログラムの読出	×
F3	SAVE	プログラムの保存	×
F4	LIST	プログラムを表示	○
F5	RUN	プログラムを実行	○

## 改訂履歴

2019.08 初版発行

2020.05 5-5. プログラムの修正 誤字を修正

2022.01 Tello が飛ばない場合の対処法を追加