

レーダーモジュール評価キット

TITAN T14RE

ソフトウェア説明書

エスタカヤ電子工業株式会社

## もくじ

1. 概要.....	2
2. 動作概略.....	3
3. レーダー動作に関する言葉の定義.....	5
4. インターフェース説明.....	8
4.1. データインターフェース.....	8
4.1.1. フレーム設定の選択.....	8
4.1.2. データフォーマット.....	8
4.1.3. 2次元データフォーマット.....	9
4.1.4. 1次元データフォーマット.....	10
4.2. MMIC コマンドインターフェース.....	11
4.2.1. 通信設定.....	11
4.2.2. コマンド通信.....	11
4.2.3. Cfg ファイルの送信.....	12
4.3. デバイス設定インターフェース.....	13
4.3.1. 通信設定.....	13
4.3.2. コマンド通信.....	13
5. インターフェースの検索.....	15
5.1. Windows.....	15
5.2. Linux.....	18

## 1. 概要

TITAN T14RE レーダーモジュール(以下、本機)との通信は USB を使用します。USB 通信により動作設定を行い、データを取得します。通信の目的ごとに、データ用、MMIC コマンド用、デバイス設定用の 3 つのインターフェースを持ちます。このため、ホストからは 3 つの USB デバイスとして認識されます。

USB デバイスとしての仕様は以下です。

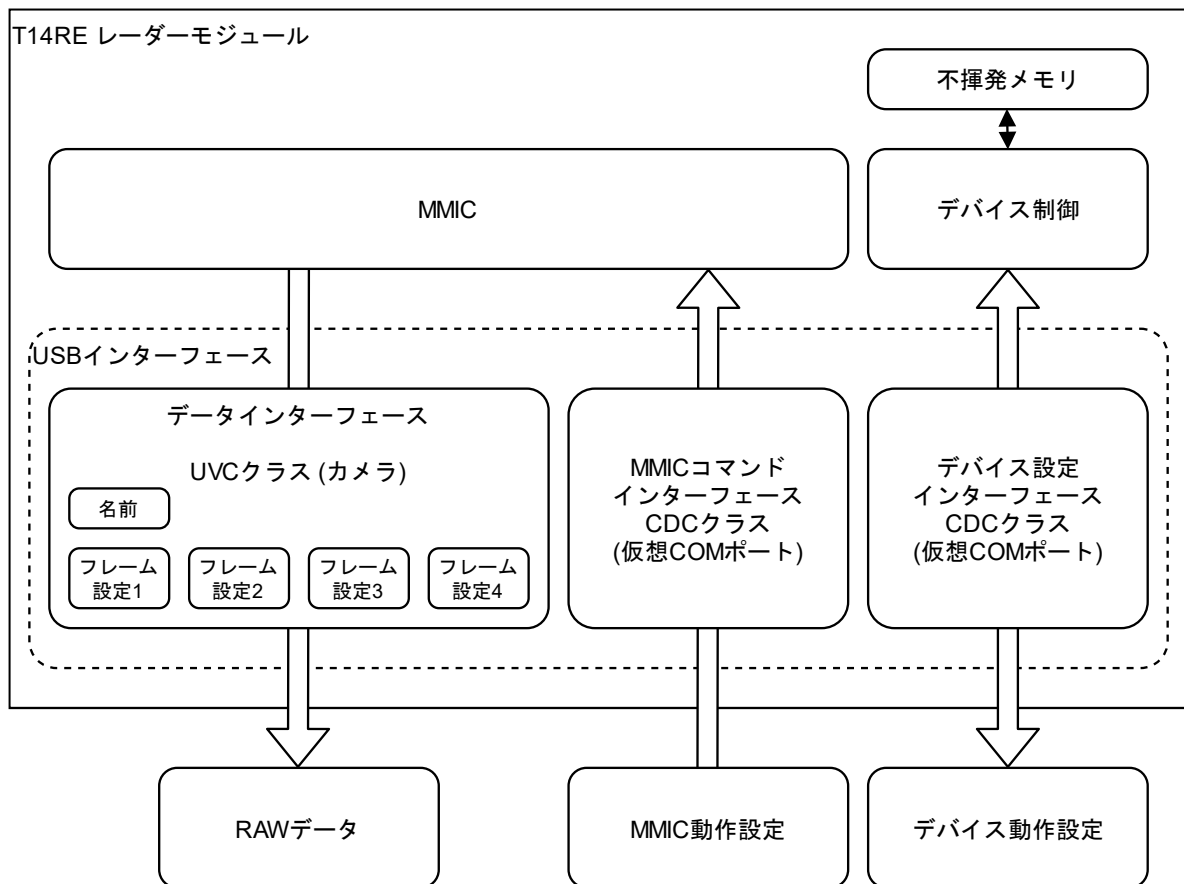
項目	値
ベンダーID	0x1364
プロダクト ID	0x802B
シリアル	デバイスの名前 (出荷時はデバイスのシリアル)
メーカー名	S-TAKAYA ELECTRONICS INDUSTRY CO. LTD.
デバイス名称	ST219-0033
データインターフェース	
デバイスクラス	UVC (カメラ)
インターフェース番号	0
デバイス名称	ST219-0033(Data):デバイス名称
MMIC コマンドインターフェース	
デバイスクラス	CDC (仮想 COM ポート)
インターフェース番号	2
デバイス名称	ST219-0033(Command):デバイス名称
デバイス設定インターフェース	
デバイスクラス	CDC (仮想 COM ポート)
インターフェース番号	4
デバイス名称	ST219-0033(Control):デバイス名称

USB デバイスとしてのシリアルはデバイスの名前として変更することが可能です。1 つのホストに複数接続して使用する場合は、このデバイスの名前を利用して実デバイスと各インターフェースの対応付けが可能になります。ただし、1 つのホストに同時に同じ名前のものを接続しないでください。

各インターフェースは標準の USB デバイスクラスとして定義されていますので、各 OS の標準のドライバを使用して通信を行うことができます。

## 2. 動作概略

下図は本機の各ブロックとインターフェースの関係とデータの流についてのブロック図です。



本機がホストに接続されると、不揮発メモリから設定を読み出し、その内容に従って動作を規定し、ホストに USB デバイスとしての情報を返します。ホストはこの情報を受取ることで、本機を USB デバイスとして認識し、USB 通信が可能になります。

不揮発メモリによって記憶されているもので、USB デバイスとしての動作を規定しているものは、デバイスの名前と、フレーム設定(カメラとして扱うことができる画像サイズの設定)です。

デバイスの名前は複数の本機を 1 つのホストで使用する場合の対応付けに使用できるものです。出荷時はデバイスのシリアル番号と同じですが、デバイス設定用通信インターフェースを使用して変更することが可能です。ただし、同じ名前のものを同時に接続すると正しく動作させることができません。

フレーム設定はデータインターフェースが扱うことができるデータのサイズです。データインターフェースは異なる 4 つのデータサイズを扱うことができます。出荷時は標準の Cfg ファイルのデータが転送できるような設定になっています。異なるデータサイズの転送が必要になる場合は、デバイス設定インターフェースを使用してフレーム設定を変更し、デバイスを再起動することで、指定のデータサイズの転送が可能になります。

本機を動作させるためにはレーダーとしての動作設定が必要です。このために MMIC コマンドインターフェースを使用して、あらかじめ用意されている Cfg ファイルの各行のコマンドを送信します。

データを取得するにはデータインターフェースを使用します。データを取得する前にどのフレーム設定を使用するか選択する必要があります。デバイスのフレーム設定を取得する方法として、事前に Cfg ファイルの設定に対応したフレーム

設定を把握しておく方法と、デバイスからフレーム設定を取得して、その中から適合するフレーム設定を検索する方法があります。デバイスから取得する方法では、データインターフェースから取得する方法と、デバイス設定インターフェースから取得する方法があります。後者の方がより詳細な情報が取得可能です。詳細についてはデータフォーマットの説明をご参照ください。実際に転送されるデータサイズは Cfg ファイルの内容によります。標準の各 Cfg ファイルのデータサイズについては Cfg 説明書.pdf をご参照ください。

フレーム設定の選択後、データインターフェースからデータを取得することが可能です。

### 3. レーダー動作に関する言葉の定義

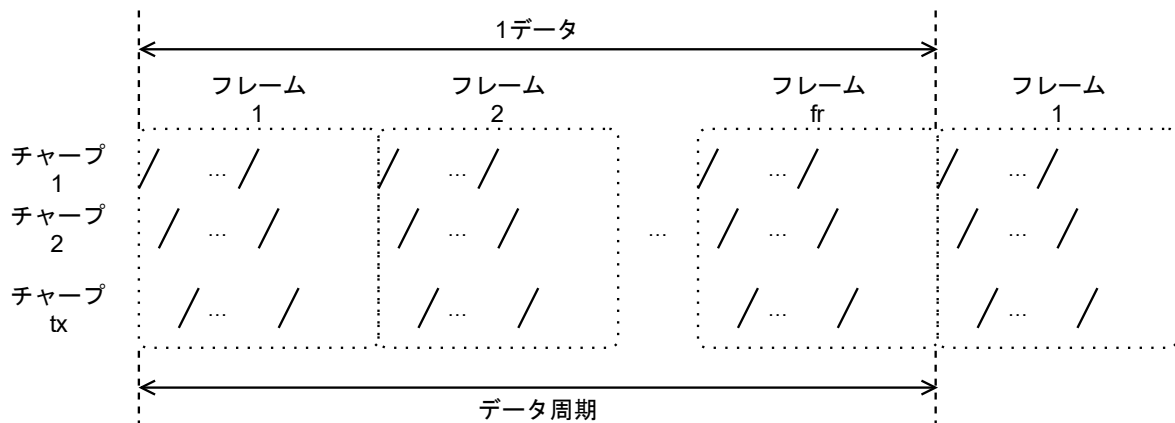
本機の動作説明を行う前に、レーダー動作に関する言葉の定義の説明を行います。チャープなど、一般的な言葉については、ここでの定義と異なることもあるかもしれませんが、本機についてはここでの定義に基づいて説明を行いますこと、ご了承ください。

- データ、データ周期、データレート

データはホストへ RAW データ、FFT データを転送する際の転送単位で、1 以上のフレームの繰り返しで構成されます。データとデータ間に最終フレームに含まれる待機期間以外の待機期間はありません。

データ周期を 1 データの時間とし、データレートを 1 秒間あたりのデータ転送回数とします。データレートはデータ周期の逆数です。

1 データに含まれるフレーム数は変更することが可能です。



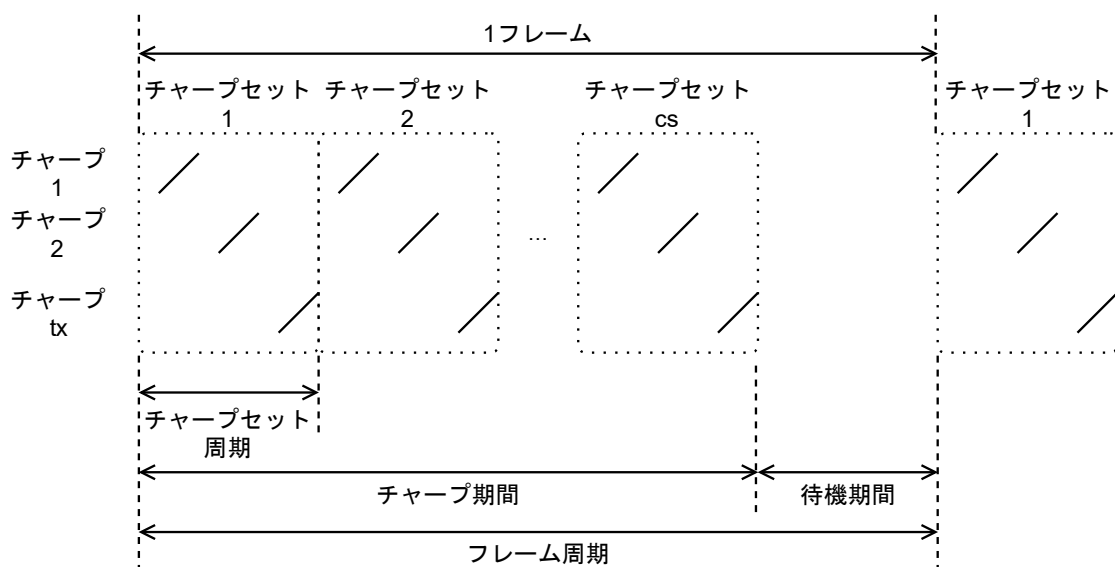
- フレーム、チャープセット周期、チャープ期間、待機期間、フレーム周期

フレームは速度成分の解析単位で、1 以上のチャープセットの繰り返しとその後の待機期間で構成されます。

1 フレームのうち、チャープセットの繰り返し期間をチャープ期間とします。また、1 チャープセットの時間をチャープセット周期とします。速度成分の解析において、チャープ期間は速度分解能に、チャープセット周期は最大速度に影響します。

フレーム周期をチャープの期間と待機期間を合わせた 1 フレームの時間とします。

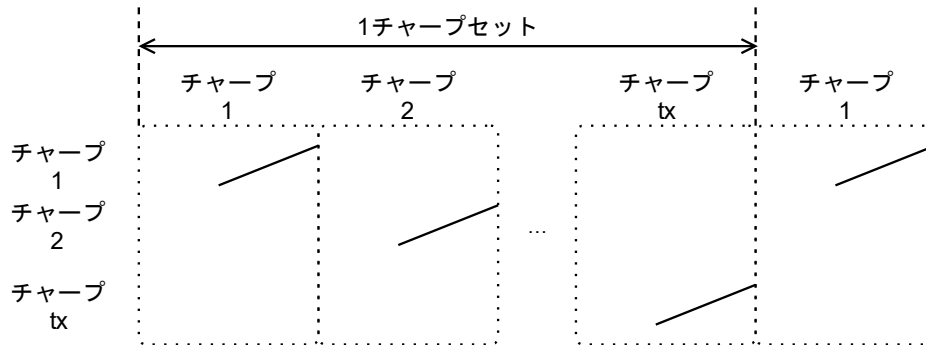
1 フレームに含まれるチャープセットの数、フレーム周期は変更することが可能です。



- チャープセット、チャープ

チャープセットはチャープの繰り返し単位で、1以上のチャープで構成されます。チャープ毎に電波を送信するアンテナを選択することができます。チャープセットとチャープセット間に次のチャープに含まれる送信休止期間以外の待機期間はありません。

1 チャープセットに含まれるチャープ数、各チャープでの送信アンテナは変更することが可能です(同時に複数のアンテナからの送信は技適の申請内容に逸脱するためできません)。



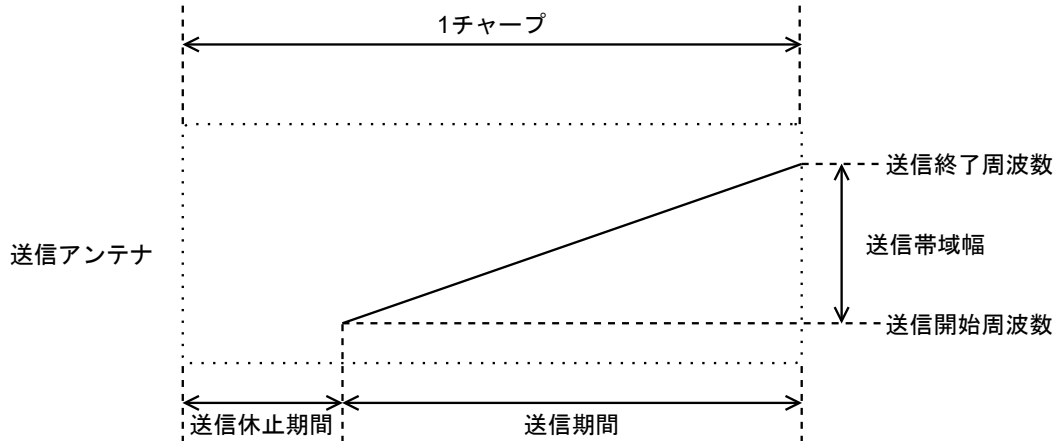
- チャープ、送信休止期間、送信期間、送信開始周波数、送信終了周波数、送信帯域幅

チャープは電波の送信タイミングの最小単位で、送信休止期間と送信期間で構成されます。

送信開始周波数を電波の送信開始時の周波数とし、送信終了周波数を送信終了時の周波数とします。また、送信帯域幅を送信期間に変化した周波数とします。

送信期間、送信開始周波数、送信終了周波数の変更は技適の申請内容に逸脱するためできません。

送信休止期間は変更することが可能です。

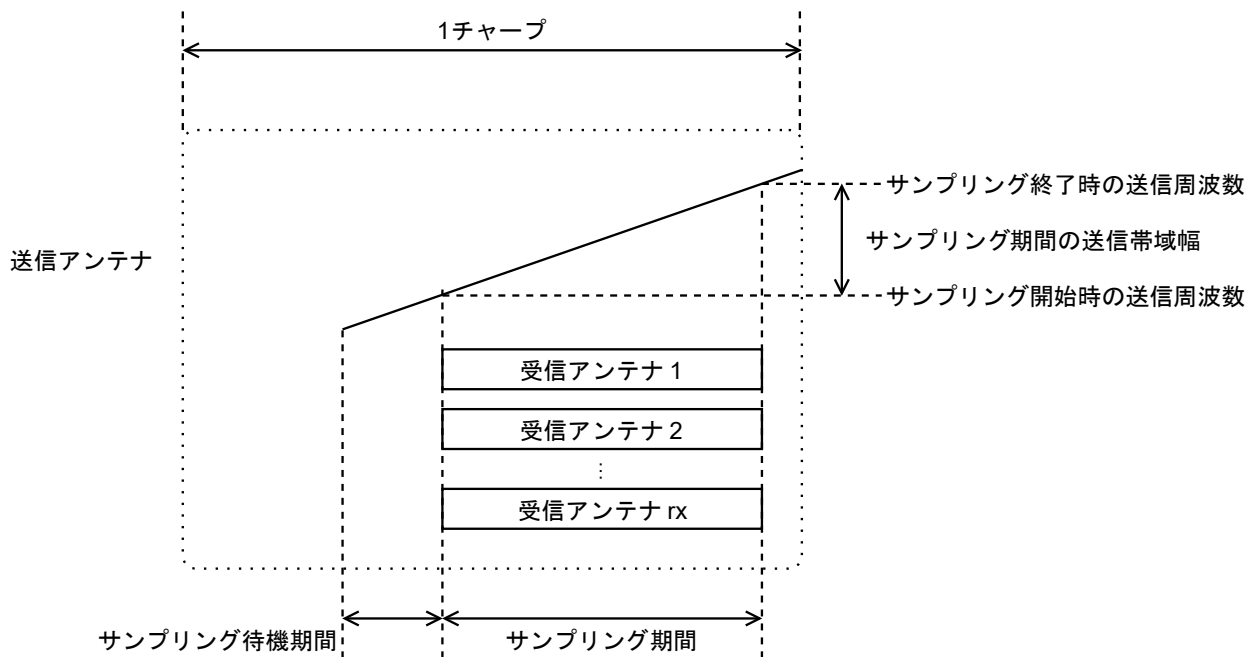


- サンプル数、サンプリング周波数、サンプリング期間、サンプリング期間の送信帯域幅

チャープ期間中に受信が有効になっている受信アンテナから IF 信号(IQ データ)の取り込みを行います。取り込みはチャープ開始後、サンプリング待機期間後から開始されます。取り込みは指定のサンプリング周波数で、サンプル数の取り込みで終了します。したがってサンプリング期間は、サンプル数をサンプリング周波数で割ったものになります。また、サンプリング期間の送信帯域幅を、サンプリング期間中の送信電波の周波数変化とします。

サンプリング期間の送信帯域幅は距離分解能に影響します。サンプル数は最大距離に影響します。

サンプル数、サンプリング周波数、サンプリング待機期間、有効受信アンテナは変更することが可能です。





## 4. インターフェース説明

### 4.1. データインターフェース

本機からデータを取得するためのインターフェースです。USB の UVC デバイスクラスとして認識されますので、各 OS 標準のドライバを使用して通信が可能です。

データの取得や、フレーム設定の選択のための API として、カメラ画像取り込み用のものが使用できます。Windows 向けには弊社からも API を提供しています。Windows 以外の OS を使用される場合は、OpenCV 等を使用してください。

データはカメラ画像として転送されます。カメラ画像としてみた場合のデータフォーマットは YUY2 です。正しくデータを取得するためにはデータを YUY2 のまま扱う必要があります。一部の OS、バックエンド、API の組合せでは自動的に RGB に変換するものがありますので、そのような環境では正しくデータを取り込むことができません。具体的には Windows + Direct Show + OpenCV です。Windows で OpenCV を使用してデータを取得される場合は、バックエンドとして Microsoft Media Foundation を使用してください。

#### 4.1.1. フレーム設定の選択

データインターフェースからデータを取得するためには、データサイズに適したフレーム設定を選択する必要があります。選択方法は 2 種類あり、1 つはフレーム設定の番号を指定する方法です。弊社の API ではこの方法で指定を行います。一方で、画像の幅と、高さを指定する方法があります。

データはカメラ画像として転送されるため、幅と高さの情報を持っています。幅は受信アンテナ数とサンプル数と 2 を掛けた値です。高さはフレーム数とチャープセット数とチャープ数を掛けた値です。画像の幅と高さを指定する方法では、使用する Cfg ファイルからこの 2 つの値を算出し、転送する画像サイズを選択することでフレーム設定の選択を行います。画像フォーマット、フレームレートの選択も行えますが、通常選択する必要はありません。

#### 4.1.2. データフォーマット

使用する API によって、データフォーマットは画像データフォーマット、もしくは、1 次元データフォーマットになります。画像データフォーマットは、画像のようにデータに幅と高さの概念があり、2 次元の配列で表されるフォーマットです。一方、1 次元データフォーマットは、画像データフォーマットに幅、高さの概念がなく、画像の左上から右下へ向かって走査したデータです。

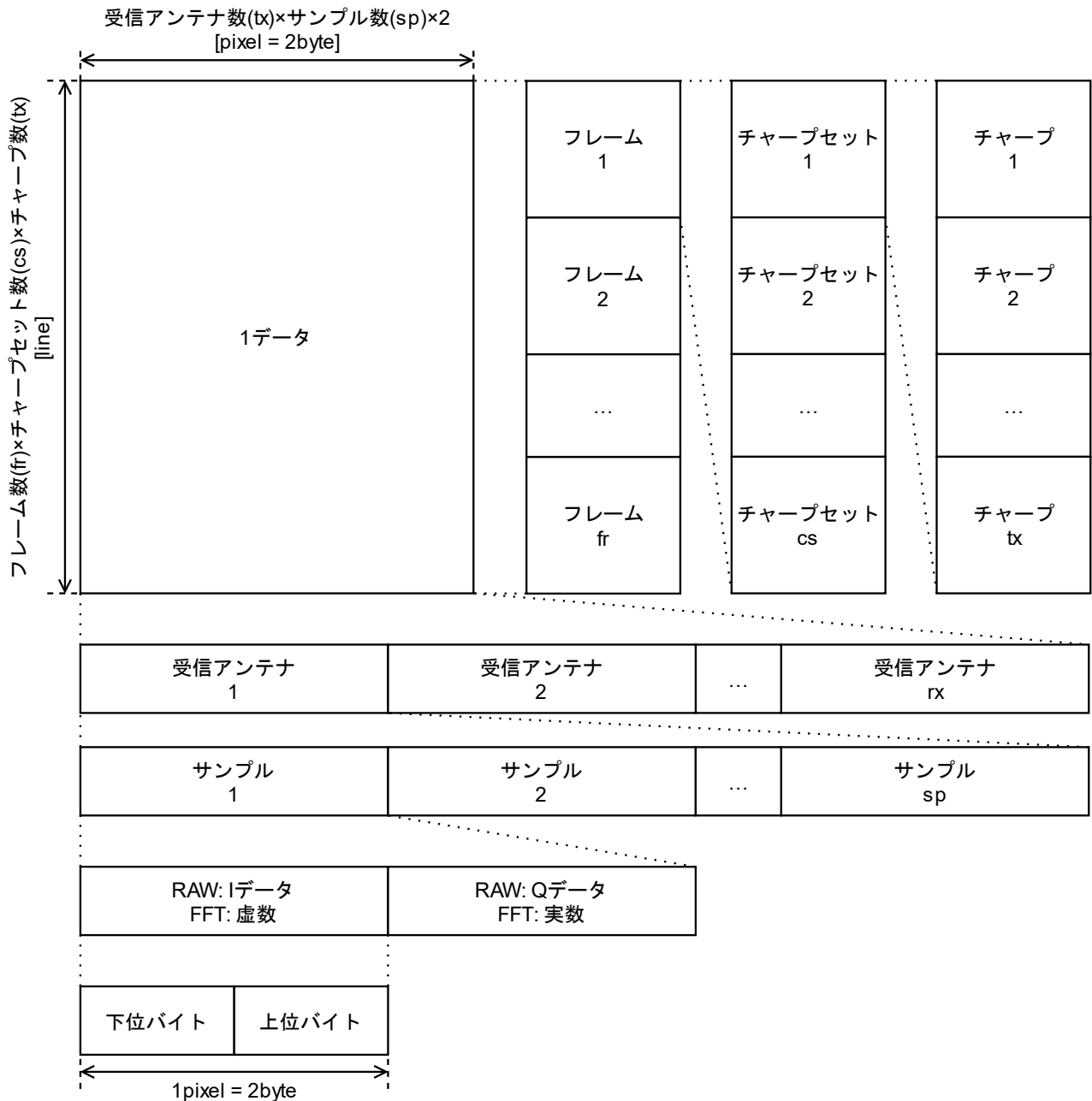
#### 4.1.3. 2次元データフォーマット

YUY2 画像としてエンコードされたデータです。

幅は受信アンテナ数とサンプル数と 2 を掛けた値です。高さはフレーム数とチャープセット数とチャープ数を掛けた値です。

1 画素としては YU、もしくは YV の 2 バイトですが、YUYV の 4 バイト単位でデコードを行います。前半の YU が I データ、もしくは虚数で、後半の YV が Q データ、もしくは実数です。どちらも Y を下位、U、V を上位として符号付 16bit 整数としてデコードします。

フレーム、チャープセット、チャープ、受信アンテナ、サンプルの並びは下図をご参照ください。

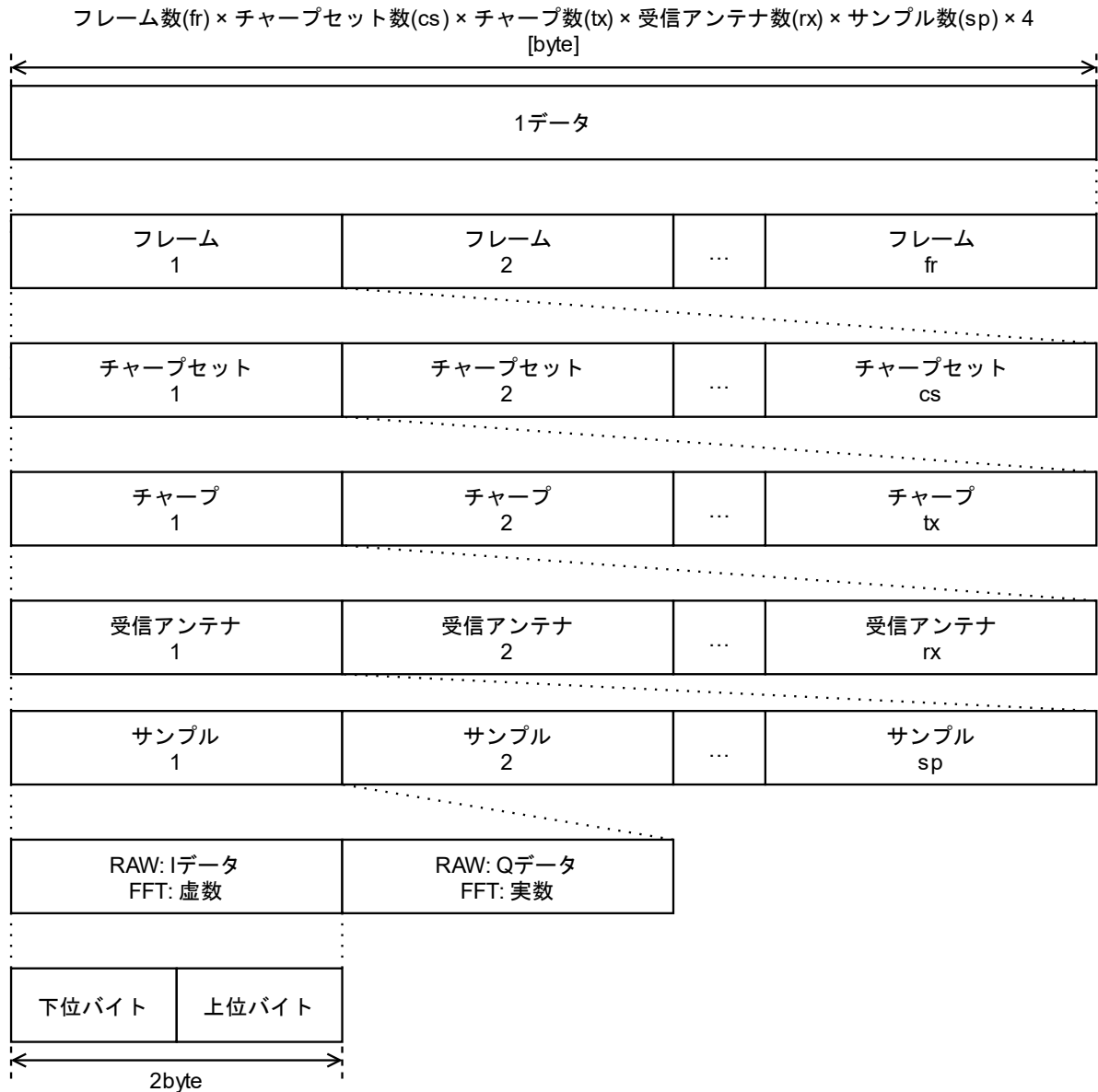


#### 4.1.4.1 次元データフォーマット

エンコードなしのデータです。

デコードは4バイト単位で行います。前半の2バイトがIデータ、もしくは実数で、後半の2バイトがQデータ、もしくは虚数です。どちらもリトルエンディアンで符号付16bit整数としてデコードします。弊社のAPIではshort配列として扱っていますので、配列の先頭からIデータ、Qデータ、Iデータ、Qデータ、…、もしくは、虚数、実数、虚数、実数、…の並びです。

フレーム、チャープセット、チャープ、受信アンテナ、サンプルの並びは下図をご参照ください。



## 4.2. MMIC コマンドインターフェース

本機の MMIC の動作設定を行うためのインターフェースです。USB の CDC デバイスクラスとして認識されますので、各 OS 標準のドライバを使用して通信が可能です。

送信や受信のための API として、各開発プラットフォーム標準のものが使用できます。Windows 向けには弊社からも API を提供しています。Windows 以外の OS を使用される場合は、各開発プラットフォーム標準のものを使用してください。例えば Python であれば pyserial が使用できます。

### 4.2.1. 通信設定

インターフェースの通信設定は下記です。

項目	値
ボーレート	115,200bps
データビット	8bit
ストップビット	1bit
パリティ	なし
ハンドシェイク	なし
フォーマット	コマンドライン
改行コード	LF
プロンプト	mmwDemo:/>

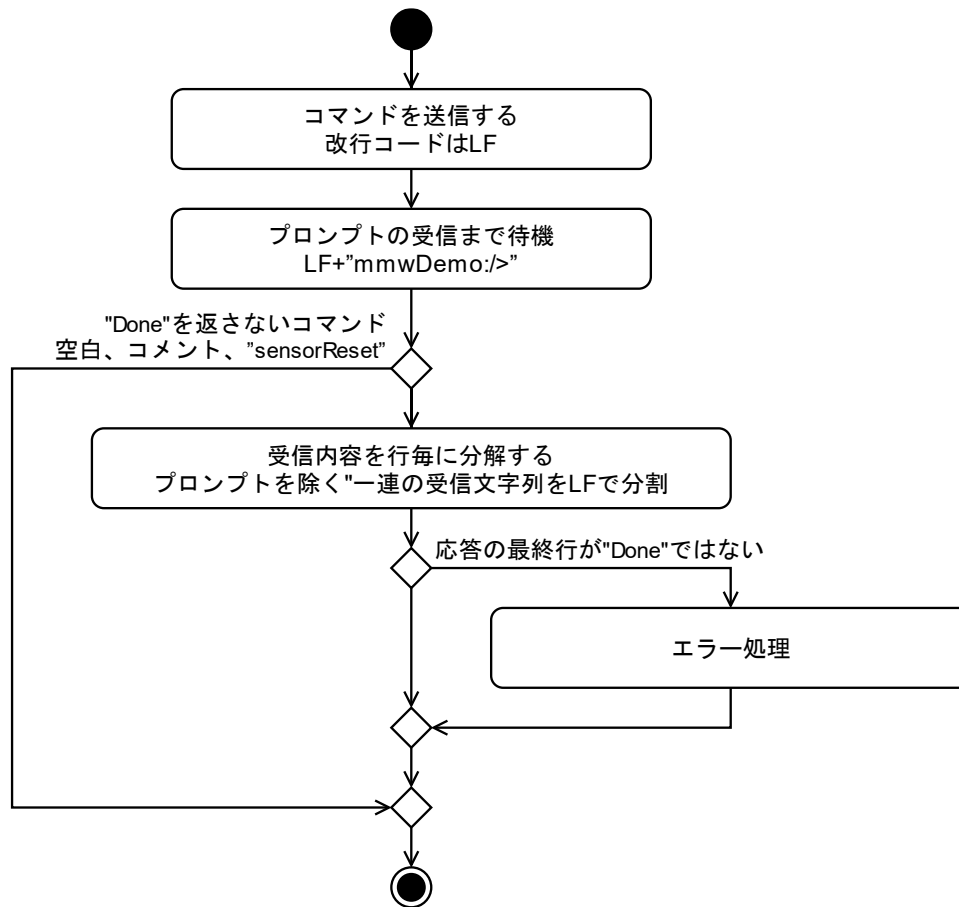
### 4.2.2. コマンド通信

ホスト側の送信 1 文字毎に MMIC へ送信され、MMIC によってエコーバックされます。コマンドは改行コードで受理され、解釈、実行されます。結果の応答はコマンドごとに異なりますが、一部のコマンドを除き、成功した場合は最終行で“Done”と応答し、その後プロンプトを返します。プロンプトが返れば、次のコマンドが送信可能です。

エラーではないのに“Done”を返さないコマンドは以下です。

- 空白行  
0 個以上の空白だけのコマンドはプロンプトだけを返します。
- コメント行  
‘%’ から始まるコメント行は“Skipped”とプロンプトを返します。
- “sensorReset”コマンド  
“sensorReset”コマンドは MMIC で改行コード受信後に再起動が行われるため、改行コードの応答を含めてその後応答が不定になり、再起動完了で再起動後のメッセージとプロンプトを返します。

以下はコマンド通信のフロー図です。



#### 4.2.3. Cfg ファイルの送信

Cfg ファイルの各行を先頭から 1 行ごとにファイルの最後まで送信します。1 行送信ごとに応答の確認を推奨します。応答の確認を行わない場合はプロンプトが返ってきていることを保障できるだけ待機してから、次行の送信を行ってください。

### 4.3. デバイス設定インターフェース

本機のデバイス設定を行うためのインターフェースです。USB の CDC デバイスクラスとして認識されますので、各 OS 標準のドライバを使用して通信が可能です。

送信や受信のための API として、各開発プラットフォーム標準のものが使用できます。Windows 向けには弊社からも API を提供しています。Windows 以外の OS を使用される場合は、各開発プラットフォーム標準のものを使用してください。例えば Python であれば pyserial が使用できます。

デバイス設定インターフェースはデバイスの設定が主であり、データを取得するために必ずしも必要ではありません。

#### 4.3.1. 通信設定

インターフェースの通信設定は下記です。

項目	値
ボーレート	115,200bps
データビット	8bit
ストップビット	1bit
パリティ	なし
ハンドシェイク	なし
フォーマット	コマンドライン
改行コード	LF
プロンプト	>

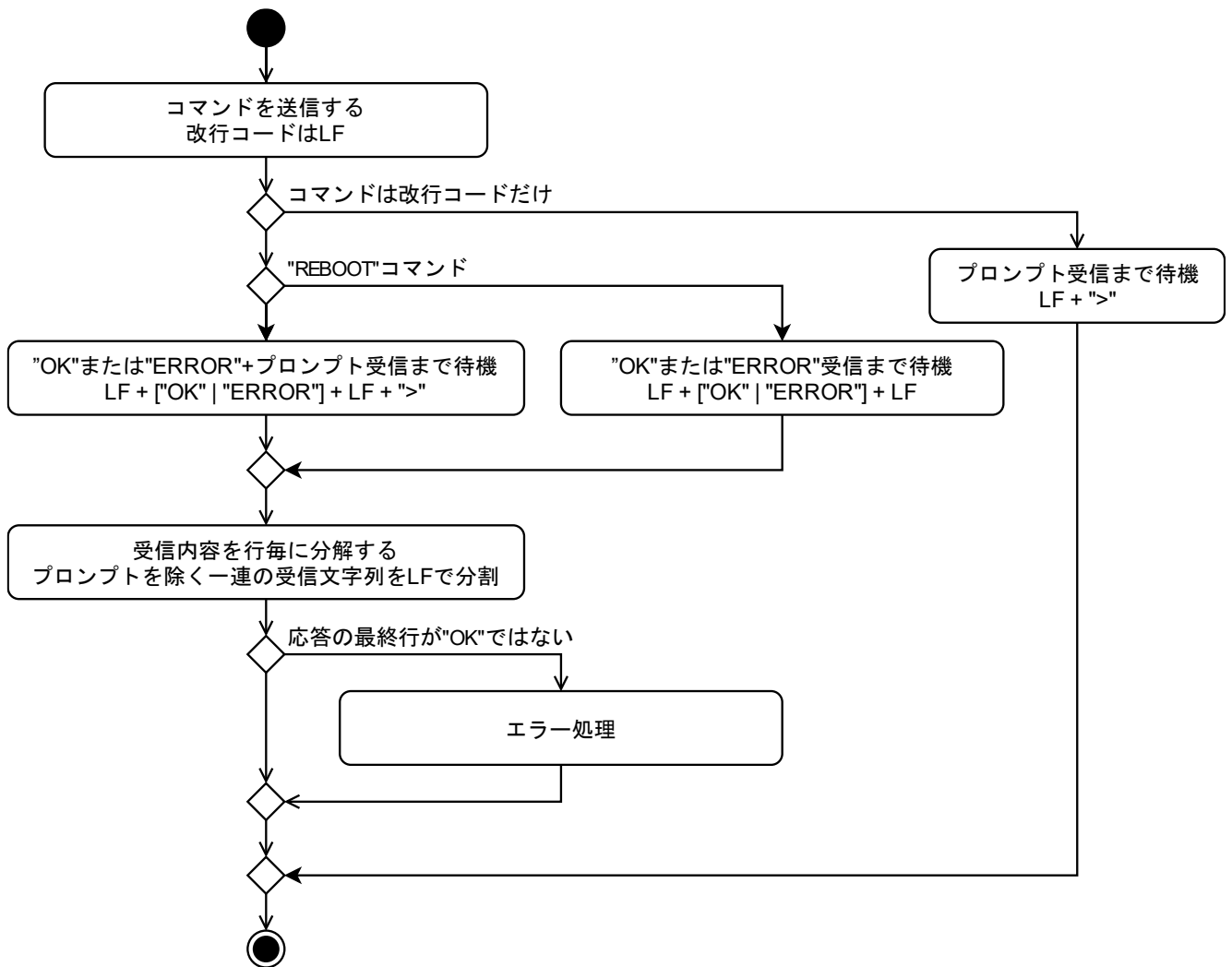
#### 4.3.2. コマンド通信

ホスト側の送信 1 文字毎にデバイスへ送信され、デバイスによってエコーバックされます。コマンドは改行コードで受理され、解釈、実行されます。結果の応答はコマンドごとに異なりますが、一部のコマンドを除き、成功した場合は最終行で“OK”、失敗した場合は“ERROR”と応答し、その後プロンプトを返します。プロンプトが返れば、次のコマンドが送信可能です。

“OK”、“ERROR”、またはプロンプトを返さないコマンドは以下です。

- 改行コード  
改行コードだけのコマンドはプロンプトだけを返します。
- “REBOOT”コマンド  
“REBOOT”コマンドはデバイスが改行コード受信後に“OK”を応答した後に再起動が行われるため、この後プロンプトを返しません。再起動により本機は一旦切断され、再起動完了後に再度 USB デバイスとして使用可能になります。

以下はコマンド通信のフロー図です。



## 5. インターフェースの検索

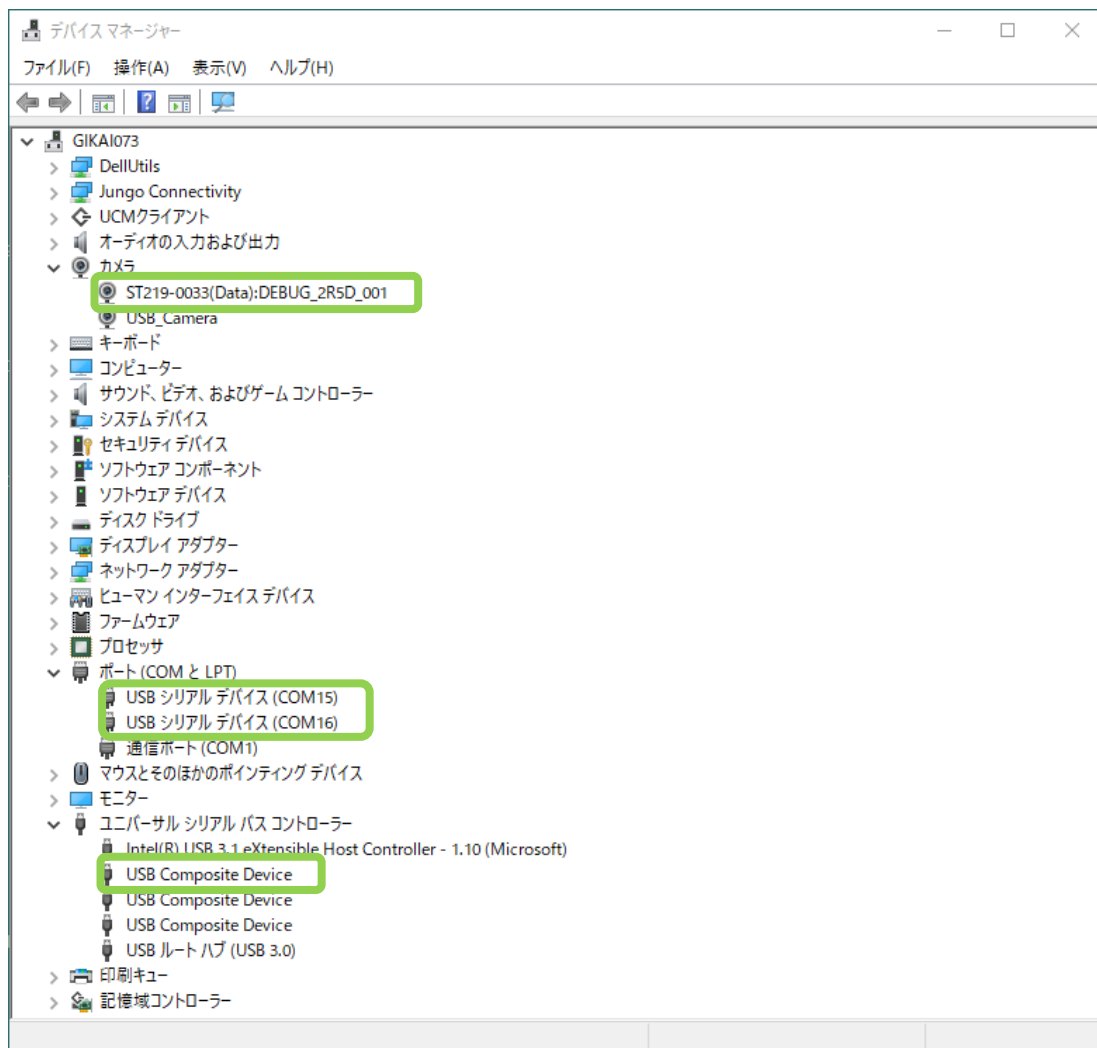
本機と通信を行うためのインターフェースの検索方法について Windows の場合と Linux の場合で説明します。

### 5.1. Windows

弊社から提供する API、もしくはアプリの T14reTool を使用することで、対応するインターフェース、および、使用するために必要な各種情報を取得することができます。

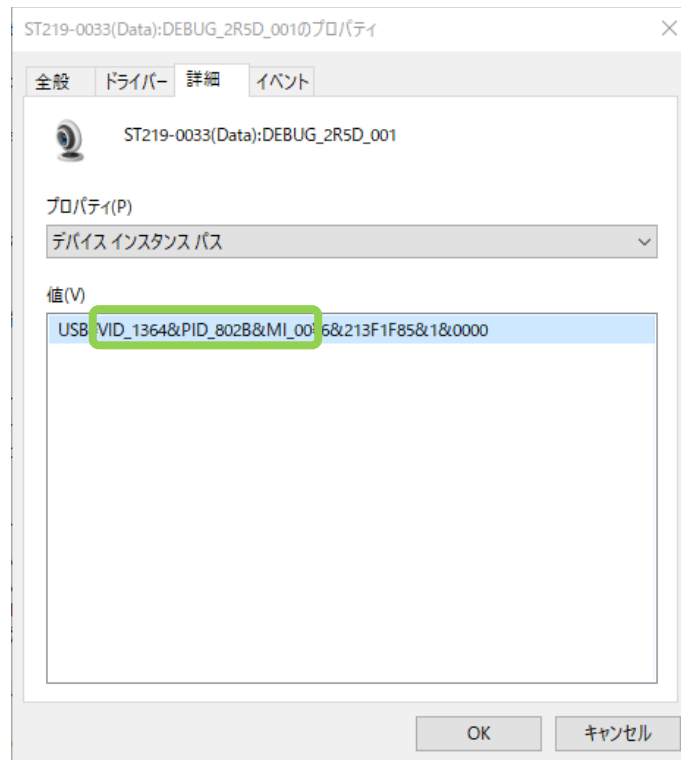
```
D:\¥T14RE>t14retool -deviceInfo
DeviceIndex, DeviceName, DataIfFriendlyName, DataIfCaptureDeviceIndex, CommandIfPortName, ControlIfPortName, UsbSpeed
0, DEBUG_2R5D_001, ST219-0033(Data):DEBUG_2R5D_001, 0, COM15, COM16, 3
```

本機は複数の機能を持つ USB Composite Device として OS に認識されます。USB Composite Device のクラスは USB です。USB Composite Device を親として、Camera クラスにデータインターフェース、Ports クラスに MMIC コマンドインターフェースとデバイス設定インターフェースが列挙されます。

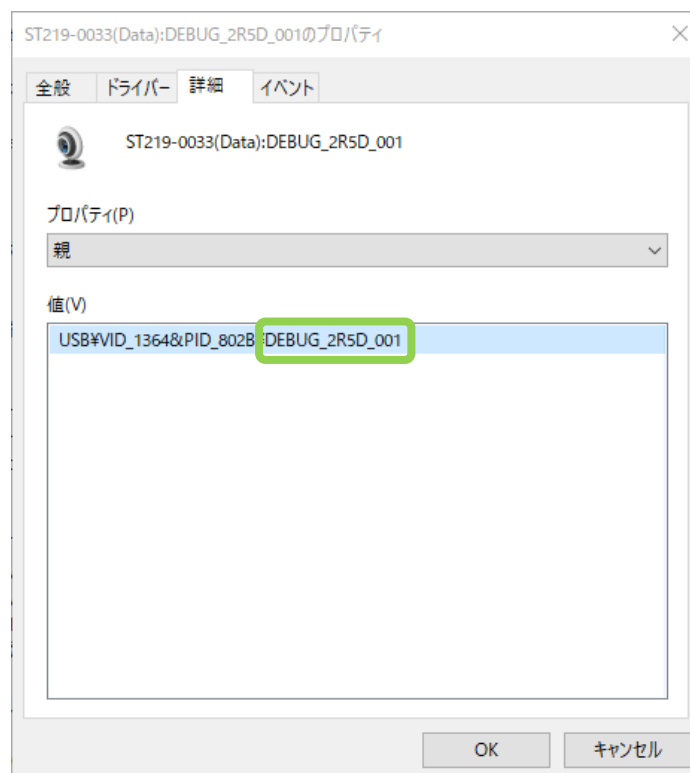




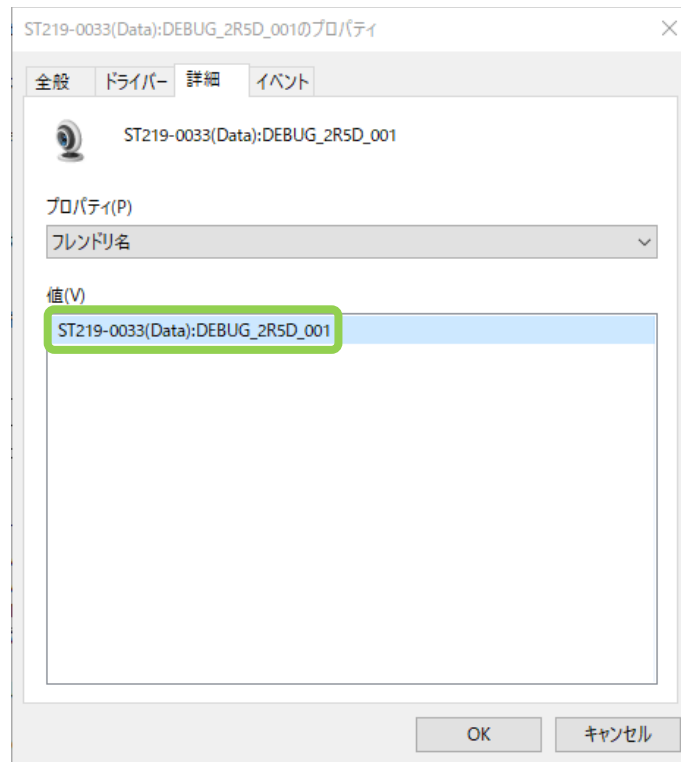
接続している本機が 1 台の場合は、各インターフェースのデバイスインスタンスパスに含まれる、ベンダーID、プロダクトID、インターフェース番号により各インターフェースを識別できます。



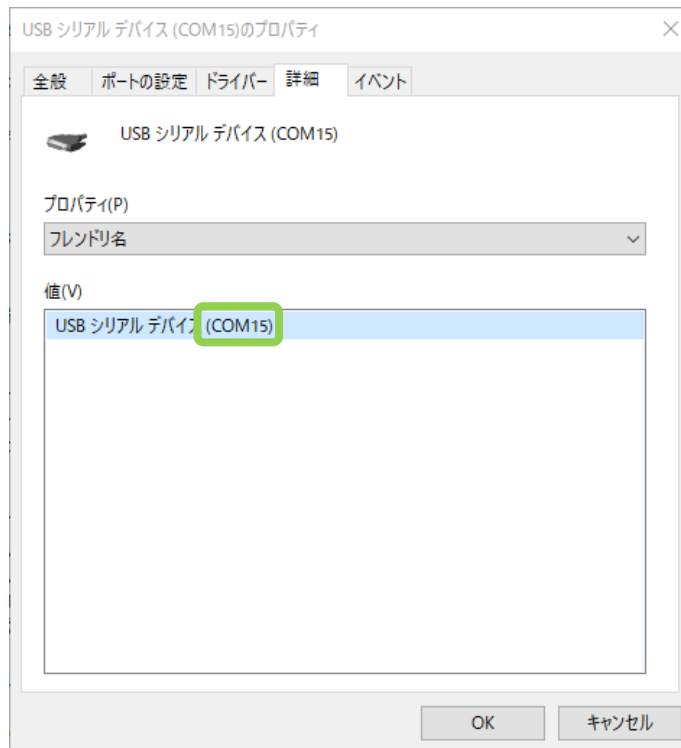
複数接続している場合は、各インターフェースの親にデバイスの名前が含まれますので、このことにより識別が可能です。



データインターフェースを使用するためには API によって、フレンドリ名かキャプチャデバイスとしてのインデックスを指定する必要があります。フレンドリ名はインターフェースのフレンドリ名から取得できますが、インデックスはバックエンドが列挙する中から何番目に列挙されたかを検索する必要があります。



MMIC コマンドインターフェース、デバイス設定インターフェースを使用するためには、COM ポート名を指定する必要があります。COM ポート名はインターフェースのフレンドリ名の括弧の中に含まれますので、ここから取得可能です。



## 5.2. Linux

Linux 向けには弊社から API やアプリの提供はありませんが、標準コマンドを使用して検索することが可能です。

データインターフェースを使用するためには UVC クラス用のドライバである V4L2 が必要です。ご使用の環境にインストールされていない場合は、あらかじめインストールをお願いします。本機を接続すると/dev/video?(?は数字)として認識されます。下記のコマンドを実行して、どのデバイスを使用すればよいのか検索します。

```
linaro@linaro-alip:~/Documents$ ls -l /dev/v4l/by-id
total 0
lrwxrwxrwx 1 root root 12 Mar 29 02:30 usb-S-TAKAYA_ELECTRONICS_INDUSTRY_CO._LTD._ST219-0033_DEBUG_2R5D_001-video-index0 -> ../../video5
lrwxrwxrwx 1 root root 12 Mar 29 02:30 usb-S-TAKAYA_ELECTRONICS_INDUSTRY_CO._LTD._ST219-0033_DEBUG_2R5D_001-video-index1 -> ../../video6
```

この例のデバイスの名前は“DEBUG\_2R5D\_001”です。デバイス名称である“ST219-0033”と合わせて本機であることが認識できます。/dev/video5 と/dev/video6 の 2 つありますが、使用するのは index0 が示す/dev/video5 です。

MMIC コマンドインターフェース、デバイス設定インターフェースは/dev/ttyACM?(?は数字)として認識されます。下記のコマンドを実行して、どのデバイスを使用すればよいのか検索します。

```
linaro@linaro-alip:~/Documents$ ls -l /dev/serial/by-id
total 0
lrwxrwxrwx 1 root root 13 Mar 29 02:30 usb-S-TAKAYA_ELECTRONICS_INDUSTRY_CO._LTD._ST219-0033_DEBUG_2R5D_001-if02 -> ../../ttyACM0
lrwxrwxrwx 1 root root 13 Mar 29 02:30 usb-S-TAKAYA_ELECTRONICS_INDUSTRY_CO._LTD._ST219-0033_DEBUG_2R5D_001-if04 -> ../../ttyACM1
```

/dev/ttyACM0 と/dev/ttyACM1 の 2 つあります。インターフェース番号 2 の if02 が示す/dev/ttyACM0 が MMIC コマンドインターフェースで、インターフェース番号 4 の if04 が示す/dev/ttyACM1 が、デバイス設定インターフェースです。

シリアル通信を行うにはデバイスに対して読み込みと書き込みの権限が必要です。権限がない場合は、権限を与えてください。

## 変更履歴

日付	Rev	担当	内容
2022/4/4	1.0	田口誠	初回作成
2022/6/6	1.1	田口誠	FFT データの並びを修正
			以上