

## 目次

1	この文書について.....	1
1.1	記載内容.....	1
1.2	内容の変更について.....	2
1.3	免責事項.....	2
2	conanair 自動モードのコンポーネントと動作の特徴.....	2
2.1	コンポーネント .....	2
2.2	動作の特徴 .....	3
2.3	制約事項.....	4
2.4	conanair の識別.....	4
3	ファイル・パス表記について .....	4
4	設定ファイル.....	5
4.1	設定内容の記述に関する共通ルール .....	5
4.2	設定内容の反映.....	5
4.3	接続鍵ファイル .....	5
4.4	ID・Tag 関連付けファイル .....	6
4.5	測定スケジュール設定ファイル .....	6
4.5.1	繰り返し指定 .....	7
4.5.2	短い時間間隔のスケジュールの扱い.....	8
4.5.3	デフォルト設定に関する制約事項.....	8
4.5.4	スケジュール管理の特例.....	8
4.6	APP_ROOT 設定ファイル .....	8
5	測定結果ファイル.....	10
5.1	測定結果の伝達経路.....	10
5.2	フォルダーツリー .....	10
5.3	データフォーマット .....	10
5.3.1	要約情報.....	11
5.3.2	実データ .....	13
	改訂履歴 .....	14

### 1 この文書について

#### 1.1 記載内容

この文書では、自動モードで動作する conanair を利用するアプリケーションを作成する際に必要となる情報を記載します。主に当該アプリケーションが conanair とやり取りする

データについて記載しますが、その理解に必要なとなる背景情報についても記載します。

## 1.2 内容の変更について

現在 conanair は開発の過程にあり、その途上で得られた重大な知見を反映して仕様を変更する開発スタイルを採っています。この文書に記載する範囲の仕様も変更対象に含まれます。この文書は、記載された文書日付時点の最新仕様のスナップショットです。

この文書に記載される範囲を含むすべての仕様は製品のリリースにより確定します。最初の製品がリリースされた後は、仕様の変更は一般的なりリース管理に従います。

## 1.3 免責事項

前節に記載した通り、最初の製品がリリースされるまでは、この文書に記載された内容は既に変更されている可能性があります。重要な事業上の決定をこの文書の内容に基づき行われる場合、変更がないか弊社まで問い合わせ・ご確認されるようお願いいたします。お問い合わせ・ご確認無くこの文書に基づいて重要な事業上の決定を行い、実際の仕様が既に変更されていたことにより損害等が発生した場合、弊社はその一切の責を負いません。

また弊社の事業上の判断により、この文書に記載された製品の機能の一部、または全部、あるいは製品そのもののリリースを中止する場合があります。これらがリリースされなかったことにより損害等が発生した場合でも、特段の約定がある場合を除き、弊社はその一切の責を負いません。

# 2 conanair 自動モードのコンポーネントと動作の特徴

## 2.1 コンポーネント

conanair 自動モードは以下のコンポーネントで構成されます。

- conanair
  - WiFi 接続振動センサーです
  - 自動モードをサポートするファームウェアが搭載されている必要があります
- 基地局
  - conanair からデータを受け取り、動作指示を行う一連のアプリケーション群です
  - 専用のコンピューター、または他の用途用に既に存在しているサーバー等の上で稼働します
  - Windows および Linux オペレーティングシステム上で稼働可能です
  - 常時連続稼働する必要があります
  - 自動的に時刻合わせのできる環境で稼働する必要があります（「conanair ネットワーク」も参照）
- conanair ネットワーク
  - conanair が接続する WiFi ネットワークが必要です

- 同一ネットワーク（ディレクテッド・ブロードキャスト・パケットの到達範囲）に複数の conanair を接続可能で、多くの場合最大数はネットワークの制限で決まります
- 基地局はこの WiFi ネットワークに参加するか、あるいは WiFi ネットワークと通信可能な有線 LAN に接続します
- 同一ネットワークに複数の基地局を接続することは可能ですが、通常アクティブ/スタンバイなど、ある時点で通信する基地局が 1 つであるような構成にします
- 複数の基地局がデータを共有する仕組みを設けることにより、ある時点で複数の基地局が通信可能な構成を実現できます
  - ☆ これは論理的可能性であり、具体的な設計などは準備されていません
- 他の用途のネットワークを共用する、または専用のネットワークを構築する、そのどちらも可能です
  - ☆ 要求されるセキュリティレベルなどに応じて決定します
- 孤立した専用のネットワークを構築するのが最も容易に高セキュリティを実現できますが、この場合基地局の時刻合わせのために特別な措置が必要になります
  - ☆ 措置の一例は、基地局を 2 つのネットワークインターフェイスを備えたコンピュータ上で稼働させ、1 つを conanair ネットワーク、他方を NTP サーバーと通信できるネットワークに接続することです
  - ☆ 措置のもう一つの例は、基地局が稼働するコンピュータに電波時計など、本質的に正確な時計（Time Source）を装備することですが、これは一般に高価です

## 2.2 動作の特徴

conanair は電池による長期間動作を実現するため大半を Deep-Sleep 状態で過ごし、所定の時刻に復帰して測定を行い、その結果を基地局に送信します。

- Deep-Sleep 中は、強制的な復帰および電池の脱着によるリセット以外の操作を受け付けません
- 1 回に Deep-Sleep できる時間は概ね 1 時間なので、所定の時刻になるまで復帰して直ちに Deep-Sleep 状態に移行すること繰り返します
- 前項の動作のため、強制的な復帰を行っても直ちに Deep-Sleep 状態に移行してしまうため、Deep-Sleep を中断するには、実質的に電池の脱着によるリセット以外に方法がありません
- Deep-Sleep からの自動的な復帰を制御するタイマーは、最大数%の時間の誤差があります
  - 大半は個体差で、これは校正処理により補正することが可能です
  - 個体差を補正した後に残る誤差は最大 0.5%程度と見込まれます
    - ☆ 誤差による測定時刻のズレが許容範囲に収まるよう、測定頻度を決める必要

があります

- 結果を送信する際、基地局の時計により次回測定タイミングを補正するため、誤差が累積することはありません

## 2.3 制約事項

前節記載の特徴のため、遠隔操作による測定スケジュールの変更は、変更前スケジュールによる測定が行われるまで反映されません。

測定スケジュールを直ちに反映するには conanair に対して物理的な操作、つまり電池の脱着によりリセットを行う必要があります。

## 2.4 conanair の識別

conanair はハードウェア毎にユニークな ID と任意に設定できる Tag 名で識別します。

- ID ハードウェア製造時に決定され、変更できません
- Tag 名は利用者が任意に設定できます
  - ある時点で複数の conanair に同じ Tag 名を設定することはできません
  - conanair のハードウェアを交換した場合など、以前の Tag 名を引き継ぐことができます
- 設定ファイルの 1 つで ID と Tag 名の対応関係を定義します
  - 適正な ID と Tag 名の対応関係を保つのはユーザーの責任です
- ID でなければ識別できない場合を除き、設定およびデータは Tag 名で識別します

## 3 ファイル・パス表記について

これ以降、ファイル・パスは以下のように表記します。

- パスの区切り文字は / (スラッシュ) を用います
- 基地局が動作しているコンピューター上の、conanair に関するツリーの最上位パスを「CAIR\_ROOT」と表現します
  - 具体的な場所は、基地局アプリをインストールしたフォルダー内の CAIR\_ROOT サブフォルダーです
- conanair の ID を「CAIR\_ID」と表現します
- conanair の Tag 名を「CAIR\_TAG」と表現します
- アプリが動作しているコンピューター上の、conanair に関するツリーの最上位パスを「APP\_ROOT」と表現します
  - 具体的な場所は、基地局の設定ファイルの一つに記述します
- ファイル・パスの大文字小文字を区別する/しないは OS に依存します
  - 区別しない前提で重複しないようネーミングを決めています
  - 区別する OS では本文書記載通りでないと正常に動作しません

## 4 設定ファイル

### 4.1 設定内容の記述に関する共通ルール

- レコード: 1 行が 1 レコードです。レコードは複数行にまたがることはできません
- エンコーディングは UTF-8 とします。BOM はあっても無くても構いません
- 改行文字: CR (0x0d)、LF (0x0a)、および CRLF (0x0d 0x0a)のいずれかです
- 無効行: 1 つでも有効な設定と解釈できない行が存在する場合、そのファイルに含まれる全ての設定はエラーとして扱います
- 空白行: 設定内容に影響ありません。任意の場所に任意の数の空白行を記述することができます
- 区切り文字: カンマ
- 先頭・末尾のホワイトスペース: カンマで区切った後の各フィールドの、先頭および末尾のスペースおよびタブは無視します
- コメント: 各行の # 記号、およびそれより右側はコメントとして扱います
- 余分なフィールド: 規定数を超えるフィールドが記述された場合、余分なフィールドはコメントとして扱います

### 4.2 設定内容の反映

設定を変更しても、その内容が直ちに反映されるわけではありません。

- スケジュールによる反映: 毎時 15 分および 45 分に現在の設定ファイルに記述された内容を読み込みます
  - 他のプロセスが排他的に開いている、などにより読めないファイル、およびエラーの発生したファイルの内容は反映しません
  - 前項の事象が発生した場合は、その旨メッセージを表示、またはログに記録します
- 再起動による反映: 新しい設定を直ちに反映する場合、基地局のプロセスを再起動します
  - 再起動に長時間 (10 秒以上) かけると、conanair から送られてくる結果を取りこぼす場合があります
  - 一つでも読めなかったりエラーが発生したりしたファイルがあると、基地局プロセスの再起動は失敗します

### 4.3 接続鍵ファイル

- 解説
  - これは conanair を利用するアプリで取り扱うことは、原則ありません
  - conanair と基地局が通信を開始する際、最初に相互に認証するのに使う基地局側鍵ファイルです

- conanair 初期設定操作の中で作成され、利用者が所定の場所にコピーします
  - ☆ この時 conanair 側鍵も作成され、不揮発性メモリーに保存されます
- 再作成することが可能で、この場合以前に作成した鍵ファイルは無効になります
- 有効な鍵ファイルが無いと conanair と基地局は通信できません
- 設定ファイル名と場所: CAIR\_ROOT/connection\_keys/CAIR\_ID.key
- 記述ルール: 自動生成されるファイルなので記述ルールは公開しません
- 内容のサンプル: 自動生成されるファイルなのでサンプルはありません

#### 4.4 ID・Tag 関連付けファイル

- 解説
  - これは利用者がテキストエディターなどで編集することを念頭に置いていますが、conanair を利用するアプリが編集機能を提供することも可能です
  - 有効な鍵ファイルがあれば conanair と基地局は通信できますが、このファイルに有効な記述が無い場合受け取ったデータを保存しません
- 設定ファイル名と場所: CAIR\_ROOT/tags.conf
- 記述ルール
  - フィールド 1: Tag 名
  - フィールド 2: ID
- 重複レコード:
  - 同じ Tag 名または ID が 2 回以上登場した場合はエラーになります
- 内容のサンプル:

```
# Sample tags file
P-123, cna_fa3c5e
```

#### 4.5 測定スケジュール設定ファイル

- 解説
  - これは利用者がテキストエディターなどで編集することを念頭に置いていますが、conanair を利用するアプリが編集機能を提供することも可能です
- 設定ファイル名と場所: CAIR\_ROOT/schedule.conf
- 記述ルール
  - フィールド 1: Tag 名
  - フィールド 2: スケジュールの種類 (サンプル参照)
  - フィールド 3~12: スケジュール (サンプル参照)
  - 時刻と日付のタイムゾーンは基地局が動作しているコンピューターの設定に従います
- 重複レコード:
  - 同じ Tag 名が 2 回以上登場した場合、最後のレコードが有効になります

- フィールドの順番
  - 時刻の順に並んでいる必要はありません
- 複数のフィールドが同じ値
  - その値のフィールドが 1 回だけ記述されたものとみなします
- 内容のサンプル:

```
# Sample schedule file
*, daily, 09:30, 10:30, 11:30, 12:30, 13:30, 14:30, 15:30, 16:30, 17:30, 18:30
P-123, daily, 09:30, 13:30, , , , , , ,
P-456, weekly, Mon 09:30, Wed 09:30, Fri 09:30, , , , , ,
C-567, monthly, Mon1 09:30, Mon2 09:30, Mon3 09:30, Mon4 09:30, , , , ,
C-657, monthly, 05 09:30, 15 09:30, 25 09:30, , , , , ,
P-234, daily, 09:00,+01:30*5, , , , , , ,
```

- 備考:
  - ✧ Tag 名「\*」はデフォルト設定 (Tag 名による個別の設定が無い場合に使用される)
  - ✧ 時刻は 24 時間制
  - ✧ 「Mon1 09:30」は第 1 月曜の 9:30
  - ✧ 「05 09:30」は 5 日の 9:30
  - ✧ 第 x 曜日形式と日形式は混在可能 (非推奨)
  - ✧ 「hh:mm,+hh:mm\*n」は繰り返し指定 (詳細後述)

#### 4.5.1 繰り返し指定

- 解説
  - 具体的な時刻を羅列するのに替えて「開始時刻、+時間間隔\*繰り返し回数」を指定することができます
  - ただし既に開始時刻を過ぎている場合、「開始時刻、+時間間隔\*繰り返し回数」に合致する次のタイミングに測定を実施します
  - 次節「短い時間間隔のスケジュールの扱い」に記載した制約から、時間間隔は 30 分の倍数でなければなりません
  - スケジュールの種類が weekly の場合、開始時刻に曜日を、時間間隔に日数を含まなければなりません。なお monthly では繰り返し指定を使えません
  - スケジュールの種類が daily の場合、23:59 を超えた場合は繰り返しせず、翌日のスケジュールに従います
  - 同様に weekly の場合 日曜の 23:59 を超えて繰り返しません
  - 具体的な時刻を羅列する方法と混在させることはできません
- 繰り返し数のワイルドカードについて
  - 具体的な数値に替えてワイルドカード「?」を指定することができます

- 繰り返し数に「?」が指定された場合、スケジュールの種類の範囲内で繰り返しを続けます

#### 4.5.2 短い時間間隔のスケジュールの扱い

- 解説

- conanair は短い時間間隔で測定を繰り返すことを想定していません
- 具体的には、次の測定までに 30 分以上の間隔があることを想定しています
- 設定されたスケジュールが、毎時ゼロ分、または 30 分以外になってしまう場合は不正な設定として、その行全体を無視します
- ただし、タイマーの誤差により実際の測定間隔が結果的に 30 分未満になってしまう場合はあり得ます

#### 4.5.3 デフォルト設定に関する制約事項

- 背景

- 多数のセンサーにデフォルト設定が適用される可能性があります
- その場合何も対策しないと多くのセンサーが同時に通信を行うことになり、処理しきれない恐れがあります

- 制約事項

- デフォルト設定では、時刻および時間間隔の指定は 30 分の倍数に限定します
- つまり分に指定できるのは 00 または 30 だけで、それ以外はエラーになります

#### 4.5.4 スケジュール管理の特例

- 全てのセンサーは、ID・Tag 関連付けファイル記載順に応じて±10 秒の倍数のオフセット時間を割り当てます
  - 最初に記載された ID・Tag のオフセットはゼロ
  - 2 番目のオフセットは+10 秒
  - 3 番目のオフセットは-10 秒
  - 4 番目のオフセットは+20 秒…
- n 番目のオフセット[秒]は以下の式に従います
  - n が奇数の場合:  $-10 \times (n - 1) / 2$
  - n が偶数の場合:  $+10 \times n / 2$
- 個別設定およびデフォルト設定の別なく、センサー毎に割り当てられたオフセット時間だけずらしたタイミングで動作するよう制御されます

#### 4.6 APP\_ROOT 設定ファイル

- 解説

- これは利用者がテキストエディターなどで編集することを念頭に置いています、



conanair を利用するアプリが編集機能を提供することも可能です

➤ このファイルには以下の設定を記述します

- ✧ APP\_ROOT のパス (場所)
- ✧ 保持する測定データの世代数
- ✧ データ同期インターバル
- ✧ その他 (隠し設定)

● 設定ファイル名と場所: CAIR\_ROOT/app\_root.conf

● 記述ルール

- フィールド 1: 設定の種類 [APP\_ROOT|DATA\_GEN|DATA\_SYNC]
- フィールド 2: 設定値 (サンプル参照)

● 重複レコード:

- 同じ設定の種類が 2 回以上登場した場合、最後のレコードが有効になります

● 内容のサンプル:

```
# Sample app_root file
APP_ROOT, ¥¥server¥share¥somefolder¥subfolder¥
DATA_GEN, 3
DATA_SYNC, 10
```

➤ 備考:

- ✧ APP\_ROOT のパスは、その後ろにファイル名を結合した文字列を使ってファイルを読み書きモードで開けるものであれば良い
- ✧ DATA\_GEN の数値は保持すべき世代数
- ✧ DATA\_SYNC の数値は APP\_ROOT の下にデータを書き込む処理を行う間隔の最小値で、単位は分
- ✧ APP\_ROOT 下へのデータを書き込みは、データを受信が前回の書き込みから DATA\_SYNC 分以上経過していたら実施される

● 想定している測定データ保存処理について:

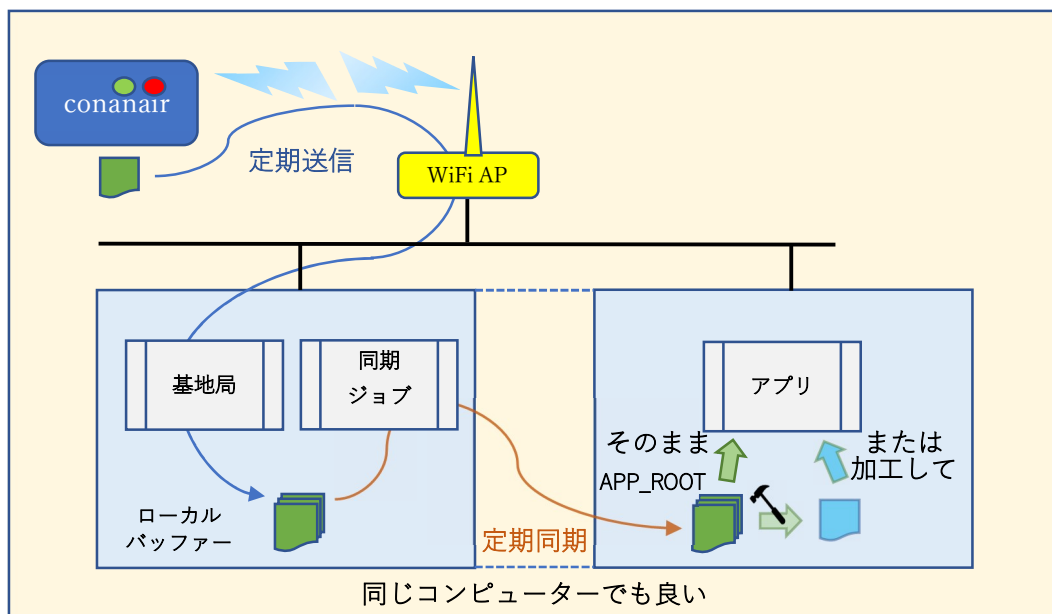
受け取った測定データを失うリスクを最小化するため、基地局アプリは以下のようにバッファの二重化を行います (次章参照)

- 測定データを受け取るプロセスは、一旦 APP\_ROOT とは異なる、基地局が稼働しているコンピューター上のストレージ内の場所にファイルを作成します
  - ✧ この場所をローカルバッファと呼びます
- 測定データを受け取るのとは別の、データを同期するプロセスが DATA\_SYNC 設定値ごとに起動し、
  - ✧ Tag 毎にローカルバッファ内に残っている測定値ファイル数を調べ、DATA\_GEN 設定値を超えている場合は古いファイルを削除します
  - ✧ APP\_ROOT とローカルバッファの差分を同期します

- ☆ 同期処理は作成されたファイルをコピーし、無くなったファイルを削除します

## 5 測定結果ファイル

## 5.1 測定結果の伝達経路

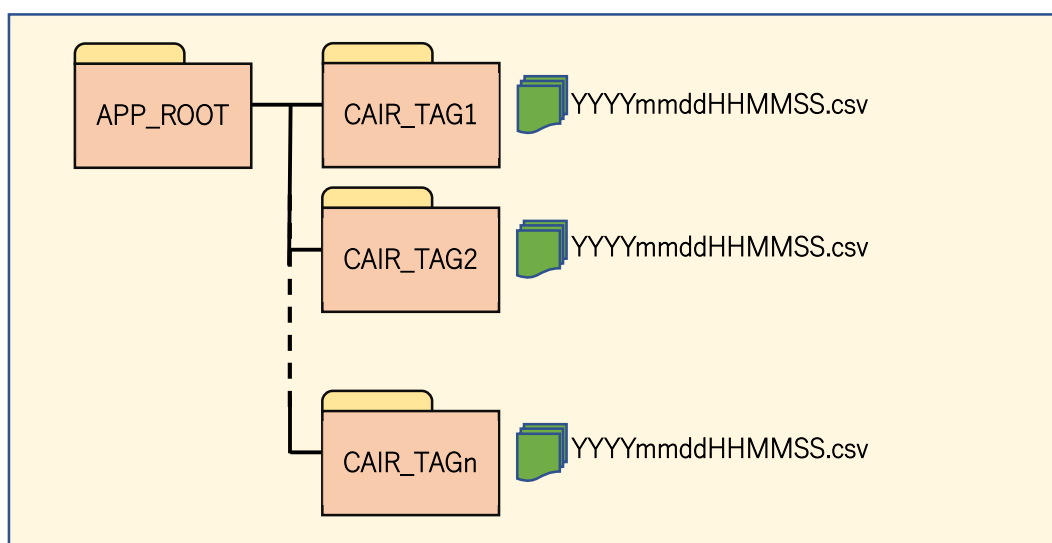


アプリが読み込むため予めデータの加工が必要な場合、アプリ側で加工の処理を用意してください。

## 5.2 フォルダーツリー

APP ROOT 下のフォルダーツリーは以下の通りです。

ローカルバッファも同じ構成ですが、アプリからのアクセスは禁止です。



## 5.3 データフォーマット

一つのファイルの中で、前半に要約情報、後半に実データが格納されます。

どちらもカンマ区切り形式ですが、列の数が異なります。

数値は全て少数を含む可能性があり、実数として解釈する必要があります。

エンコーディングおよび改行文字は以下の通りです。

- エンコーディング: BOM 付 UTF-8
- 改行文字: CR+LF (0x0d 0x0a: Windows style)

### 5.3.1 要約情報

要約情報部分の列数は 2 で、Key-Value 対を形成しています。

バージョン毎に行数は決まっていますが、変更の柔軟性を持たせるため、特別な Key により要約部分の終わりを表します。

次節に記載する実データは伝達のため非可逆的に圧縮されますが、これら要約情報は圧縮される前の値に基づき計算されているため、実データから計算されたものとは異なります。

No.	Key	Value の説明
1	ID	conanair の ID=CAIR_ID 【文字】
2	Tag	ID に関連付けられた Tag= CAIR_TAG 【文字】
3	FileName	ファイル名 (測定日時) 【文字】
4	ODR	公称サンプルレート [サンプル/秒] 【これ以降全て数値】
5	N_Samples	サンプル数 (XYZ 1 セットを一つと数える)
6	Act_ms	実際の測定時間 [ミリ秒] ● 実際のサンプルレート = $N\_Samples \times 1000 \div Act\_ms$
7	Vdd3V3	測定時の電源電圧 [V]
8	RAccAveX	各軸の、重力を含む加速度の平均値 [ $m/s^2$ ] ● センサーの、鉛直方向に対する取り付け姿勢が判る
9	RAccAveY	
10	RAccAveZ	
11	RAccPkXP	各軸の、重力を含む加速度のピーク値 [センサー読み出し値] ● 末尾の P はプラス側ピーク値 ● 末尾の N はマイナス側ピーク値 ● センサーが計測可能な最小値 -4096、最大値 4094 なので、この値が得られた場合はオーバースケールしたと推定できる (僅かながらオーバースケールしていない可能性もある)
12	RAccPkXN	
13	RAccPkYP	
14	RAccPkYN	
15	RAccPkZP	
16	RAccPkZN	

(表の続き)

No.	Key	Value の説明
17	HAccPkXP	各軸、および 3D ベクトルの重力を除外した加速度のピーク値 [m/s <sup>2</sup> ]
18	HAccPkXN	
19	HAccPkYP	● 重力加速度除去のため、F <sub>C</sub> =10Hz のハイパスフィルターを適用している
20	HAccPkYN	
21	HAccPkZP	● 末尾の P はプラス側ピーク値 ● 末尾の N はマイナス側ピーク値 ● 3D ベクトルに関しては絶対値（大きさ）なので、マイナス側は存在しない
22	HAccPkZN	
23	HAccPk3D	
24	HAccRmsX	各軸、および 3D ベクトルの重力を除外した加速度の RMS 値 [m/s <sup>2</sup> ]
25	HAccRmsY	
26	HAccRmsZ	● 重力加速度除去のため、F <sub>C</sub> =10Hz のハイパスフィルターを適用している
27	HAccRms3D	
28	VelPkXP	各軸、および 3D ベクトルの重力を除外した加速度から計算した速度のピーク値[mm/s]
29	VelPkXN	
30	VelPkYP	● 末尾の P はプラス側ピーク値 ● 末尾の N はマイナス側ピーク値 ● 3D ベクトルに関しては絶対値（大きさ）なので、マイナス側は存在しない
31	VelPkYN	
32	VelPkZP	
33	VelPkZN	
34	VelPk3D	
35	VelRmsX	各軸、および 3D ベクトルの重力を除外した加速度から計算した速度の RMS 値[mm/s]
36	VelRmsY	
37	VelRmsZ	
38	VelRms3D	
39	AP_RSSI	接続した Wi-Fi アクセスポイントの通信強度[dBm]
40	Dev_Tmp	内部基板の温度[°C]
41	END_Summary	要約部分の終わり。値は存在するが、意味は無い。(列数を揃えるため)

## 5.3.2 実データ

実データの各行に Key はありません。データのための以下の列を含みます。

ヘッダー行は存在しないので、行数は要約情報の N\_Samples と等しくなります。

No.	データ ID	説明
1	RAccX	各軸の、重力を含む加速度 [m/s <sup>2</sup> ] ● センサーの、鉛直方向に対する姿勢変化が判る
2	RAccY	
3	RAccZ	
4	HAccX	各軸、および 3D ベクトルの重力を除外した加速度[m/s <sup>2</sup> ] ● 姿勢変化の影響を受けない、純粋な振動成分 ● 重力加速度除去のため、F <sub>C</sub> =10Hz のハイパスフィルターを適用している
5	HAccY	
6	HAccZ	
7	HAcc3D	
8	VelX	各軸、および 3D ベクトルの重力を除外した加速度から計算した速度 [mm/s]
9	VelY	
10	VelZ	
11	Vel3D	

## 改訂履歴

2019-4-12        0.0.3 → 0.0.4

「1.1 記載内容」からリリース前文書に関する記述削除

2021-3-18        0.0.4 → 2.0.0

V2 へのバージョンアップにより 5.3.1 要約情報に 2 項目追加

(AP\_RSSI、Dev\_Tmp) を追加

以上