新型コロナウイルス接触確認機能付き

GPSトラッカー通信仕様書

　 Ver 1.0 2021.01.11

　 Ver 1.1 2021.01.20

株式会社アイルソフト

市川満之

目次

[1. Web API 3](#_Toc62111809)

[1.1. ユーザ情報登録 3](#_Toc62111810)

[1.2. ユーザ情報編集 3](#_Toc62111811)

[1.3. ユーザ情報取得 4](#_Toc62111812)

[1.4. IDパスワード送信 4](#_Toc62111813)

[1.5. ユーザ情報削除 5](#_Toc62111814)

[1.6. ログイン認証 5](#_Toc62111815)

[1.7. 認証コード発行 5](#_Toc62111816)

[1.8. 認証コード確認 6](#_Toc62111817)

[1.9. デバイス情報登録 7](#_Toc62111818)

[1.10. デバイス情報編集 7](#_Toc62111819)

[1.11. デバイス情報取得 8](#_Toc62111820)

[1.12. デバイス情報削除 8](#_Toc62111821)

[1.13. 認証コード登録 9](#_Toc62111822)

[1.14. 認証コード適用 9](#_Toc62111823)

[1.15. 学校情報登録 10](#_Toc62111824)

[1.16. 学校情報編集 10](#_Toc62111825)

[1.17. 学校情報取得 11](#_Toc62111826)

[1.18. 学校情報削除 11](#_Toc62111827)

[1.19. 陽性登録 12](#_Toc62111828)

[1.20. 陽性削除 12](#_Toc62111829)

[1.21. 濃厚接触登録 13](#_Toc62111830)

[1.22. 濃厚接触確認 13](#_Toc62111831)

[1.23. SMS認証コード発行 14](#_Toc62111832)

[1.24. SMS認証コード確認 14](#_Toc62111833)

[1.25. メール認証コード発行 15](#_Toc62111834)

[1.26. メール認証コード確認 15](#_Toc62111835)

[2. MQTT 16](#_Toc62111836)

[2.1. デバイス暗号配信 16](#_Toc62111837)

[2.2. デバイス情報配信 16](#_Toc62111838)

[2.3. 緊急通知配信 17](#_Toc62111839)

[2.4. 学校報告暗号配信 18](#_Toc62111840)

[2.5. 学校報告配信 18](#_Toc62111841)

[3. BLE 19](#_Toc62111842)

[3.1. 接触検知 19](#_Toc62111843)

[3.2. デバイス通信 19](#_Toc62111844)

# Web API

Web APIは、3000番ポートでアクセスする。

共通URL: http://ik1-407-35954.vs.sakura.ne.jp/api/v1/

## ユーザ情報登録

認証していないユーザ情報を登録する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | POST |
| URL | user |
| リクエスト | Body  {    ”Name”:”鈴木一郎”,    ”ZipCode”: ”464-0001”,    ”Prefectures”:”23”,    ”Address”:”名古屋市千種区末森通１丁目１番地１”,    ”TelephoneNumber”:”080-0000-0000”,    ”EmailAddress”:”support@example.jp”,    ”LoginID”:”Ichiro”,    ”Password”:”ic5.pq!r&!2(Sad”  } |
| レスポンス | ステータスコード:200 成功 200以外 エラー  エラー時のBodyは、プリザンターのエラー内容を返す。 |

## ユーザ情報編集

ユーザ情報を編集する。認証していない状態に戻す。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | PUT |
| URL | user |
| リクエスト | Body  {    ”Name”:”鈴木一郎”,    ”ZipCode”: ”464-0001”,    ”Prefectures”:23,    ”Address”:”名古屋市千種区末森通１丁目１番地１”,    ”TelephoneNumber”:”080-0000-0000”,    ”EmailAddress”:”support@example.jp”,    ”LoginID”:”Ichiro”,    ”Password”:”ic5.pq!r&!2(Sad”  } |
| レスポンス | ステータスコード:200 成功 200以外 エラー  エラー時のBodyは、プリザンターのエラー内容を返す。 |

## ユーザ情報取得

メールアドレスを指定して、ユーザ情報を取得する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | GET |
| URL | user |
| リクエスト | Body  {    ”EmailAddress”:”support@example.jp”  } |
| レスポンス | ステータスコード:200 成功 200以外 エラー  成功時のBody  {      ”Name”: ”鈴木一郎”,      ”ZipCode”: ”464-0001”,      ”Prefectures”: 23,      ”Address”: ”千種区末森通１丁目１８番地１”,      ”TelephoneNumber”: ”080-0000-0000”,      ”EmailAddress”: ”support@example.jp”,      ”LoginID”: ”Ichiro”,      ”Authentication”: false  }  400エラー時のBody  {    ”Message”: ”Not recordh”  }  それ以外のエラー時のBodyは、プリザンターのエラー内容を返す。 |

## IDパスワード送信

メールアドレスを指定して、ユーザ情報を取得する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | GET |
| URL | idpassword |
| リクエスト | Body  {    ”EmailAddress”:”support@example.jp”  } |
| レスポンス | ステータスコード:200 成功 200以外 エラー  400エラー時のBody  {    ”Message”: ”Not recordh”  }  それ以外のエラー時のBodyは、プリザンターのエラー内容を返す。 |

## ユーザ情報削除

ログインIDを指定して、ユーザ情報を削除する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | DELETE |
| URL | user |
| リクエスト | Body  {    ”LoginID”: ”Ichiro”  } |
| レスポンス | ステータスコード:200 成功 200以外 エラー  エラー時のBodyは、プリザンターのエラー内容を返す。 |

## ログイン認証

ログインIDとパスワードを指定して、ログインできるか確認する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | GET |
| URL | login |
| リクエスト | Body  {    ”LoginID”: ”Ichiro”,    ”Password”:”ic5.pq!r&!2(Sad”  } |
| レスポンス | ステータスコード:200 成功 200以外 エラー  400，401エラー時のBody  {    ”Message”: ”Not recordh”　メッセージは変わる  }  それ以外のエラー時のBodyは、プリザンターのエラー内容を返す。 |

## 認証コード発行

ログインIDを指定して、２段階認証の認証コードを発行する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | PATCH |
| URL | auth/request |
| リクエスト | Body  {    ”LoginID”: ”Ichiro”  } |
| レスポンス | ステータスコード:200 成功 200以外 エラー  エラー時のBodyは、プリザンターのエラー内容を返す。 |

## 認証コード確認

ログインIDと認証コードを指定して、２段階認証を確認する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | PATCH |
| URL | auth/verify |
| リクエスト | Body  {    ”LoginID”: ”Ichiro”,    ”MailCode”:”SeKab8Ll”,    ”SmsCode”: ”PhqkQYRM”  } |
| レスポンス | ステータスコード:200 成功  400 認証コード期限切れ  401 認証コードの間違い  402 一時パスワードなし  以外はプリザンターのエラー  成功時のBody  {    ”TemporaryPassword”: ”Password”（一日毎に変化する。）  }  400、401，402エラー時のBody  {    ”Message”: ”Authentication code mismatch” 　メッセージは変わる  }  上記以外のエラー時のBodyは、プリザンターのエラー内容を返す。 |

## デバイス情報登録

デバイス情報を登録する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | POST |
| URL | device |
| リクエスト | Body  {    ”ID”:”TAROU01”,    ”LoginID”: ”Ichiro”,    ”Key”:”$dcV$%sfp%PN\*(-”  } |
| レスポンス | ステータスコード:200 成功　402 一時パスワードなし  以外はプリザンターのエラー  成功時のBody  {    ”TemporaryPassword”: ”Password”（一日毎に変化します。）  }  402エラー時のBody  {    ”Message”: ”Not Temporary Password”  }  上記以外のエラー時のBodyは、プリザンターのエラー内容を返す。 |

## デバイス情報編集

デバイス情報を編集する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | PUT |
| URL | device |
| リクエスト | Body  {    ”ID”:”TAROU01”,    ”LoginID”: ”Ichiro”,    ”Key”:”$dcV$%sfp%PN\*(-”  } |
| レスポンス | ステータスコード:200 成功　402 一時パスワードなし  以外はプリザンターのエラー  成功時のBody  {    ”TemporaryPassword”: ”Password”（一日毎に変化します。）  }  402エラー時のBody  {    ”Message”: ”Not Temporary Password”  }  上記以外のエラー時のBodyは、プリザンターのエラー内容を返す。 |

## デバイス情報取得

デバイスIDを指定して、デバイス情報を取得する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | GET |
| URL | device |
| リクエスト | Body  {    ”ID”:”TAROU01”  } |
| レスポンス | ステータスコード:200 成功 200以外 エラー  成功時のBody  {    ”ID”:”TAROU01”,    ”LoginID”: ”Ichiro”,    ”Key”:”$dcV$%sfp%PN\*(-”,    ”DaysLeft”:180  }  エラー時のBodyは、プリザンターのエラー内容を返す。 |

## デバイス情報削除

デバイスIDを指定して、デバイス情報を削除する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | DELETE |
| URL | device |
| リクエスト | Body  {    ”ID”:”TAROU01”  } |
| レスポンス | ステータスコード:200 成功 200以外 エラー  エラー時のBodyは、プリザンターのエラー内容を返す。 |

## 認証コード登録

認証コードを登録する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | POST |
| URL | device/code |
| リクエスト | Body  {    ”AuthCode”:”T0zcZsseXDSwaui4oSmvu3U1aGuk77JEDx75”,    ”ValidDays”:180  } |
| レスポンス | ステータスコード:200 成功　200以外 エラー  エラー時のBodyは、プリザンターのエラー内容を返す。 |

## 認証コード適用

認証コードをデバイスに適用する。デバイスの有効期限が残っている場合は、認証コードの残り日数に加算される。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | PATCH |
| URL | device/code/apply |
| リクエスト | Body  {    “ID”:”TAROU01”,    ”AuthCode”:”T0zcZsseXDSwaui4oSmvu3U1aGuk77JEDx75”  } |
| レスポンス | ステータスコード:200 成功　401デバイスIDなし 400認証コードなし  以外はプリザンターのエラー  成功時のBody  {    ”AuthCode”:”T0zcZsseXDSwaui4oSmvu3U1aGuk77JEDx75”,    ”ValidDays”:180  }  401、402エラー時のBody  {    ”Message”: ”Not Device ID”  }  上記以外のエラー時のBodyは、プリザンターのエラー内容を返す。 |

## 学校情報登録

学校情報を登録する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | POST |
| URL | school |
| リクエスト | Body  {    ”ID”:”MARMO.CO”,    ”Key”:”$dcV$%sfp%PN\*(-”  } |
| レスポンス | ステータスコード:200 成功　402 一時パスワードなし  以外はプリザンターのエラー  成功時のBody  {    ”TemporaryPassword”: ”Password”（一日毎に変化します。）  }  402エラー時のBody  {    ”Message”: ”Not Temporary Password”  }  上記以外のエラー時のBodyは、プリザンターのエラー内容を返す。 |

## 学校情報編集

学校情報を編集する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | PUT |
| URL | school |
| リクエスト | Body  {    ”ID”:”MARMO.CO”,    ”Key”:”$dcV$%sfp%PN\*(-”  } |
| レスポンス | ステータスコード:200 成功　402 一時パスワードなし  以外はプリザンターのエラー  成功時のBody  {    ”TemporaryPassword”: ”Password”（一日毎に変化します。）  }  402エラー時のBody  {    ”Message”: ”Not Temporary Password”  }  上記以外のエラー時のBodyは、プリザンターのエラー内容を返す。 |

## 学校情報取得

学校IDを指定して、学校情報を取得する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | GET |
| URL | school |
| リクエスト | Body  {    ”ID”:”MARMO.CO”  } |
| レスポンス | ステータスコード:200 成功 200以外 エラー  成功時のBody  {    ”ID”:”MARMO.CO”,    ”Key”:”$dcV$%sfp%PN\*(-”,  }  エラー時のBodyは、プリザンターのエラー内容を返す。 |

## 学校情報削除

学校IDを指定して、学校情報を削除する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | DELETE |
| URL | school |
| リクエスト | Body  {    ”ID”:”MARMO.CO”  } |
| レスポンス | ステータスコード:200 成功 200以外 エラー  エラー時のBodyは、プリザンターのエラー内容を返す。 |

## 陽性登録

陽性者として接触情報を登録する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | POST |
| URL | positive |
| リクエスト | Body  {[    “data”:{       “Time”: 20210115091123,       “TEK”: “a07d635658f82c3c4b8fb211f1e0634”,       “ENIN”:[“5fe5bedc”, “5fe5c1ac”, …] ←144個    },    繰り返し14個  ]} |
| レスポンス | ステータスコード:200 成功　以外はプリザンターのエラー  上記以外のエラー時のBodyは、プリザンターのエラー内容を返す。 |

## 陽性削除

接触情報を削除する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | DELETE |
| URL | positive |
| リクエスト | Body  {[    “data”:{       “Time”: 20210115091123,       “TEK”: “a07d635658f82c3c4b8fb211f1e0634”,       “ENIN”:[“5fe5bedc”, “5fe5c1ac”, …] ←144個    },    繰り返し14個  ]} |
| レスポンス | ステータスコード:200 成功　以外はプリザンターのエラー  上記以外のエラー時のBodyは、プリザンターのエラー内容を返す。 |

## 濃厚接触登録

濃厚接触者として接触情報を登録する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | POST |
| URL | contact |
| リクエスト | Body  {[    “data”:{       “Time”: 20210115091123,       “TEK”: “a07d635658f82c3c4b8fb211f1e0634”,       “ENIN”:[“5fe5bedc”, “5fe5c1ac”, …] ←144個    },    繰り返し14個  ]} |
| レスポンス | ステータスコード:200 成功　以外はプリザンターのエラー  上記以外のエラー時のBodyは、プリザンターのエラー内容を返す。 |

## 濃厚接触確認

接触情報から陽性者および濃厚接触者との濃厚接触を確認する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | POST |
| URL | contact |
| リクエスト | Body  {[    “data”:{       “Time”: 20210115091123,       “RPI”: “a07d635658f82c3c4b8fb211f1e0634”    },    繰り返し 接触のデータ分  ]} |
| レスポンス | ステータスコード:200 成功　以外はプリザンターのエラー  成功時のBody （濃厚接触がない場合はからの配列となる。）  {[    “data”:{       “Time”: 20210115091123,       “Type”: 0, ←濃厚接触情報       “RPI”: “17166ffd32eaf96ce56521975f312572”    },    {       “Time”: 20210115091123,       “Type”: 1, ←陽性情報       “RPI”: “07d635658f82c3c4b8fb211f1e0634a”    },  濃厚接触者のデータ分  ]}  上記以外のエラー時のBodyは、プリザンターのエラー内容を返す。 |

## SMS認証コード発行

ログインIDを指定して、SMSの認証コードを発行する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | PATCH |
| URL | auth/request/sms |
| リクエスト | Body  {    ”LoginID”: ”Ichiro”  } |
| レスポンス | ステータスコード:200 成功 200以外 エラー  エラー時のBodyは、プリザンターのエラー内容を返す。 |

## SMS認証コード確認

ログインIDと認証コードを指定して、SMSを確認する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | PATCH |
| URL | auth/verify/sms |
| リクエスト | Body  {    ”LoginID”: ”Ichiro”,    ”SmsCode”: ”PhqkQYRM”  } |
| レスポンス | ステータスコード:200 成功  400 認証コード期限切れ  401 認証コードの間違い  402 一時パスワードなし  以外はプリザンターのエラー  400、401，402エラー時のBody  {    ”Message”: ”Authentication code mismatch”  }  上記以外のエラー時のBodyは、プリザンターのエラー内容を返す。 |

## メール認証コード発行

ログインIDを指定して、メールの認証コードを発行する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | PATCH |
| URL | auth/request/mail |
| リクエスト | Body  {    ”LoginID”: ”Ichiro”  } |
| レスポンス | ステータスコード:200 成功 200以外 エラー  エラー時のBodyは、プリザンターのエラー内容を返す。 |

## メール認証コード確認

ログインIDと認証コードを指定して、メールの認証を確認する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| メソッド | PATCH |
| URL | auth/verify/ mail |
| リクエスト | Body  {    ”LoginID”: ”Ichiro”,    ”MailCode”:”SeKab8Ll”  } |
| レスポンス | ステータスコード:200 成功  400 認証コード期限切れ  401 認証コードの間違い  402 一時パスワードなし  以外はプリザンターのエラー  400、401，402エラー時のBody  {    ”Message”: ”Authentication code mismatch”  }  上記以外のエラー時のBodyは、プリザンターのエラー内容を返す。 |

# MQTT

MQTTは、8883番ポート（TLS接続）でca.crtを使い、アクセスする。

ホスト名：http://ik1-407-35954.vs.sakura.ne.jp/

MQTTのユーザの登録は、ブローカを再起動しないといけない。ブローカは0:00に再起動する。

|  |  |  |  |
| --- | --- | --- | --- |
| 機器 | ユーザ名 | パスワード | 期間 |
| 全て | temporary | 登録した時のWeb APIで取得したパスワードを利用して接続します。BLEでの設定時にデバイスへ書込む。 | 登録から翌日０時まで |
| デバイス | デバイスID | キー | 登録の翌日０時から |
| 管理者アプリ | ログインID | パスワード |
| 学校アプリ | 学校ID | キー |

メッセージの削除は、削除したいメッセージと同じTopic、DoS、payloadが空のメッセージを配信する

## デバイス暗号配信

デバイスから暗号化ルールを配信します。

|  |  |
| --- | --- |
| 項目 | 内容 |
| 配信元 | デバイス |
| 配信単位 | １日毎　0:00 |
| 配信topic | デバイス名ID/key/時刻(※)　　 ※YYYYMMDDHHMMSS |
| 購読先 | 管理者アプリ |
| 購読topic | デバイス名ID/# |
| QoS | 2 |
| message | JSON形式  {  "type":"json",  "name":["Lat", "Lng", "Step"],  "key":"キー+暗号鍵"  }  キーは、以下のルールでハッシュ化される。   1. 時刻のDDの２文字目+1文字分、時刻を切り出す。 2. その切り出した文字をキーとして、SHA256ハッシュ化する。 3. キーは既に管理者アプリからデバイスにBLEで書き込まれているため、ハッシュ化した値をマッチングし、一致した後ろの文字列を暗号キーとして取り出す。 |
| 削除 | しない　（配信単位とブローカの再起動のタイミングが同じため） |

## デバイス情報配信

デバイスから温度、湿度、緯度経度、歩数を配信します。

|  |  |
| --- | --- |
| 項目 | 内容 |
| 配信元 | デバイス |
| 配信単位 | 管理者アプリで設定された時間毎（BLEで書き込まれる） |
| 配信topic | デバイス名ID/番号/経過時間(※)　　 ※デバイス暗号配信からの秒数 |
| 購読先 | 管理者アプリ |
| 購読topic | デバイス名ID/# （デバイス暗号配信と同じ） |
| QoS | 0 （購読topicがデバイス暗号配信と同じため、購読側はDoS 2とする） |
| message | JSON形式  {  "Lat no sec":35.16, //秒を切り捨てた緯度,  "Lng no sec":136.94, //秒を切り捨てた経度,  "TEMP":21.3, //温度  "HUM":50.1, //湿度  "Air quality":25, //空気質  "Lat":35.1666901, //緯度  "Lng":136.9441721, //経度  "Step":2501 //歩数  }  赤字はデバイス暗号配信の暗号鍵で暗号化(AES256)されます。 |
| 削除 | しない |

## 緊急通知配信

デバイスでボタンを押下した時に温度、湿度、緯度経度、歩数を配信します。

|  |  |
| --- | --- |
| 項目 | 内容 |
| 配信元 | デバイス |
| 配信単位 | デバイスでボタンが一定時間（３秒）押された時 |
| 配信topic | デバイス名/emg/10秒単位の緯度/10秒単位の経度 |
| 購読先 | 管理者アプリ |
| 購読topic | 管理者アプリのスマートフォンの緯度経度から前後10秒の緯度経度の配信を購読できるようにする。  +/emg/10秒単位の緯度/10秒単位の経度  +/emg/10秒単位の緯度/10秒単位の経度-1  +/emg/10秒単位の緯度/10秒単位の経度+1  +/emg/10秒単位の緯度-1/10秒単位の経度  +/emg/10秒単位の緯度-1/10秒単位の経度-1  +/emg/10秒単位の緯度-1/10秒単位の経度+1  +/emg/10秒単位の緯度+1/10秒単位の経度  +/emg/10秒単位の緯度+1/10秒単位の経度-1  +/emg/10秒単位の緯度+1/10秒単位の経度+1 |
| QoS | 2 |
| message | JSON形式  {  "Lat":35.1666901, //緯度  "Lng":136.9441721, //経度  "Sex":1, //性別(1:男、2:女、0:なし)  "Age":10 //年齢  } |
| 削除 | ボタンが話された時 |

## 学校報告暗号配信

管理者アプリから暗号化ルールを配信します。

|  |  |
| --- | --- |
| 項目 | 内容 |
| 配信元 | 管理者アプリ |
| 配信単位 | １日毎　0:00 |
| 配信topic | 学校ID/デバイスID/key/時刻(※)　　 ※YYYYMMDDHHMMSS |
| 購読先 | 学校アプリ |
| 購読topic | 学校ID/# |
| QoS | 2 |
| message | JSON形式  {  "type":"json",  "name":["Positive Close Contacts", "Contact Close Contacts"],  "key":"キー+暗号鍵"  }  キーは、以下のルールでハッシュ化される。   1. 時刻のDDの２文字目+1文字分、時刻を切り出す。 2. その切り出した文字をキーとして、SHA256ハッシュ化する。 3. キーは管理者アプリで入力して送られているため、ハッシュ化した値をマッチングし、一致した後ろの文字列を暗号キーとして取り出す。 |
| 削除 | しない　（配信単位とブローカの再起動のタイミングが同じため） |

## 学校報告配信

管理者アプリから濃厚接触数を配信します。

|  |  |
| --- | --- |
| 項目 | 内容 |
| 配信元 | 管理者アプリ |
| 配信単位 | 管理者アプリで送信ボタンを押下した時 |
| 配信topic | 学校ID/デバイスID/Contacts/時刻(※)　　 ※YYYYMMDDHHMMSS |
| 購読先 | 学校アプリ |
| 購読topic | 学校ID/# （学校報告暗号配信と同じ） |
| QoS | 1 （購読topicが学校報告暗号配信と同じため、購読側はDoS 2とする） |
| message | JSON形式  {  "Positive Close Contacts":2,  "Contact Close Contacts":1  }  赤字は学校報告暗号配信の暗号鍵で暗号化(AES256)されます。 |
| 削除 | しない |

# BLE

BLEは、デバイスと管理者アプリ間でデータを読み書きする通信（GATT通信）と接触を検知する接続待ち（アドバタイズ）の二つで行われる。

## 接触検知

管理者アプリまたはデバイスとの接触を接続待ち状態をスキャンすることで検知する。

|  |  |
| --- | --- |
| 項目 | 内容 |
| ペリフェラル | デバイス、管理者アプリ |
| ペリフェラルのアドバタイズ周期 | 5分、４回 |
| セントラル | デバイス、管理者アプリ |
| セントラルのスキャン | 同じRPI+AEMが３回以上続いた場合、接触と検知する。 |

データ形式

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Flags | | | Complete 16bit Service UUID | | | Service Data - 16bit UUID | | | | |
| Length | Type | Flags | Length | Type | Service UUID | Length | Type | Service Data | | |
| 0x02 | 0x01 (Flag) | 0x1A | 0x03 | 0x03 (Complete 16bit Service UUID) | 0xFD6F (Exposure Notification Service) | 0x17 | 0x16 (Service Data - 16bit UUID) | 0xFD6F (Exposure Notification Service) | 16bytes RPI*i* | 4bytes AEM*i,j* |

## デバイス通信

デバイスと管理者アプリ間でデータを読み書きする通信（GATT通信）を行う。

|  |  |
| --- | --- |
| 項目 | 内容 |
| ペリフェラル | デバイス |
| アドバタイズ周期 | 1秒間隔 電源投入後３分間 |
| セントラル | 管理者アプリ |
| スキャン | marmo設定画面が表示された時 |
| 接続要求 | marmo設定画面でデバイスアイコンをタップした時 |
| ペアリング | 最初の接続時はデバイスとペアリングする。  Bluetoothペア設定コードは最初のペアリングは123456とし、２回目以降はIDの連番5桁の２の補数を計算し、５桁の10進数とする。  87654(10)→15666(16)→EA99A(16)→960922(10) |
| ボンディング | 同じデバイスと３回目以降の接続は交換したペアリングキーを保存し、ボンディングを使用する。 |
| 設定値の読み込み(read)/書き込み(write) | UUID 51f2e511-be4e-42e2-a502-0bf3aa109855  {  "id”:"M0001", //ID  "name":"鈴木太郎", //名前(最大30バイト)  "sex":1, //性別(1:男、2:女、0:なし)  "birthday":19700101, //生年月日(４バイト)  "alert humidity":40, //アラート湿度（2バイト）  "key": "wUU,8-cZb(#S$RM", //キー(最大30バイト)  "publish interval":5, //送信の間隔（2バイト）  "expiration date":20211130, //有効期限(４バイト)  "temporary password":"e8EPSC-.gJZAMy%" //一時パスワード  } |
| TEK、ENINの 読み込み(read) | 日時(YYYYMMDDHHMMSS)、生成したTEK(16Byte) + ENIN(4Byte)１４日間分  UUID 88b9d302-1d53-4743-af14-ccb68179fa75  {[  {  "time”:" 20210115091123", //時刻  "TEK":"a07d635658f82c3c4b8fb211f1e0634",  "ENIN":["5fe5bedc", "5fe5c1ac", …] ←144個  },  繰り返し14個  ]} |
| RPI、AEMの 読み込み(read) | 日時(YYYYMMDDHHMMSS)、受信したRPI(16Byte) １４日間分  UUID b9428273-c634-491c-9e0a-f3ec17cefbc9  {[  {  "time": 20210115091123,  "RPI": "17166ffd32eaf96ce56521975f312572"  },  繰り返し受信した個数  ]} |