新型コロナウイルス接触確認機能付き

GPSトラッカーシステム仕様書

　 Ver 1.0 2020.12.10

Ver 1.1 2020.12.16

Ver 1.2 2020.12.23

Ver 1.3 2020.12.25

株式会社アイルソフト

市川満之

目次

[1. 目的 3](#_Toc59818895)

[2. 前提条件 3](#_Toc59818896)

[3. 機能概要 4](#_Toc59818897)

[3.1. 接触確認機能 4](#_Toc59818898)

[3.2. GPSトラッカー機能 9](#_Toc59818899)

[4. 基本シーケンス 15](#_Toc59818900)

[4.1. 基本機能 15](#_Toc59818901)

[4.2. 接触確認機能 16](#_Toc59818902)

[4.3. GPSトラッカー機能 18](#_Toc59818903)

# 目的

本システムは、２つの目的を持つ。

１つ目はスマートフォンを持っておらず、新型コロナウイルス接触確認アプリ（COCOA)（以降COCOAと記述）を使えない子供やお年寄りに対して、本システムのデバイス（以降「デバイス」と記述）がBLEを利用したCOCOAと同じ機能を持ち、人と人との接触したことを検知、記録する。記録を管理者のスマートフォンの本システムのアプリケーション（以降「アプリ」と記述）に移し、新型コロナウイルス感染症の陽性診断が確定した者（以下「陽性者」と記述）であることが判明した場合およびその陽性者と濃厚接触があった場合に、その管理者および本人の同意のもとで、その陽性者と一定期間内に接触が確認された者に対し通知を行う。（以降接触確認機能と記述）

２つ目はGPSトラッカーとして本デバイスから本アプリに対してLTE-Mを利用してMQTT通信で緯度経度を通知し、子供やお年寄りがどこにいるか分かるようにするためである。また、緊急時にボタンを押下することで本デバイスから近辺にいる不特定多数の本アプリに対してアラートによる緊急通知を行う。（以降GPSトラッカー機能と記述）

# 前提条件

接触確認機能は、接触確認者としての定義および接触の情報を確認できるのは14日間までとし、COCOAに準ずる。また、個人情報保護の観点からGPSトラッカー機能の名前、性別、住所、生年月日、位置情報、電話番号、メールアドレス等の特定の個人が直接識別される可能性のある情報は関係性を持たない。

GPSトラッカー機能は、互助の考えから身元保証と管理する意味で名前、電話番号、メールアドレス、パスワードを入力する。デバイスには緊急通知のために、性別、生年月日を記録する。また、デバイスを使用するためには、認証コードによる有効期限をアプリからBLEを利用して保存する必要がある。

# 機能概要

## 接触確認機能

* デバイス（写真は試作品）

他者との接触について、デバイスに1メートル以内で１５分以上接触した相手のスマートフォンのCOCOAの識別子が記録される。同様にCOCOAにもデバイスの識別子が記録される。また、デバイス同士でもデバイスの識別子が記録される。

デバイスに記録された識別子は、管理者のアプリにBLEに転送する。転送された識別子から濃厚接触したかを確認する。確認した内容は学校もしくは会社にスマホのインターネットを利用して暗号化したMQTT通信で配信することができる。デバイスおよび管理者のアプリの識別子の記録は、14日間経過後に削除される。

濃厚接触を確認

BLE（ペアリングおよびボンディング）

記録された識別子を転送

新型コロナウイルス接触確認アプリ 厚生労働省Bluetoothのアイコン, ブルートゥースのロゴ, ブルートゥース, ロゴ, アイコンを, 青Bluetoothのアイコン, ブルートゥースのロゴ, ブルートゥース, ロゴ, アイコンを, 青Bluetoothのアイコン, ブルートゥースのロゴ, ブルートゥース, ロゴ, アイコンを, 青Bluetoothのアイコン, ブルートゥースのロゴ, ブルートゥース, ロゴ, アイコンを, 青

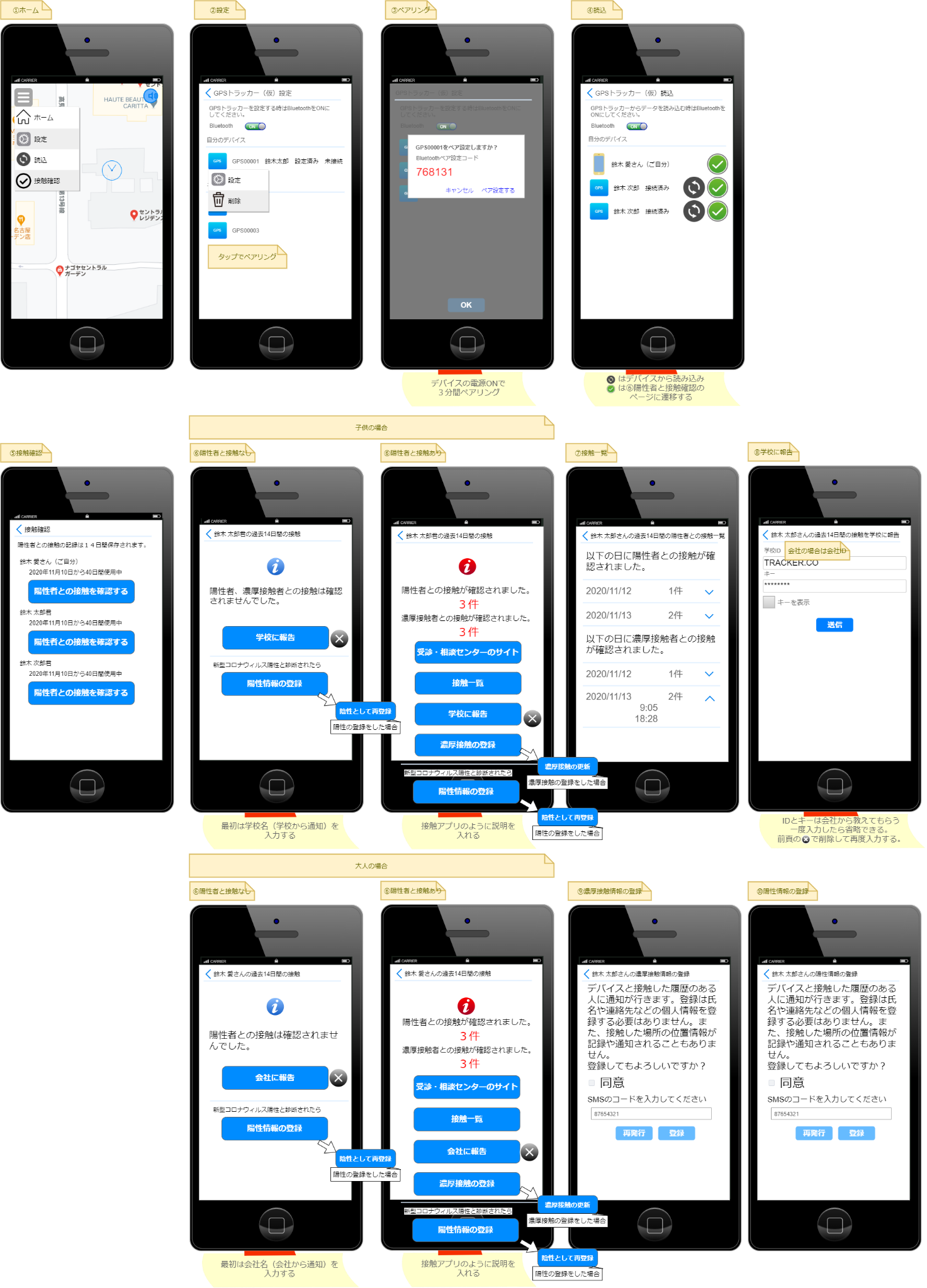
BLEで識別子を送信、記録

アプリおよびCOCOAと通信する

BLEで識別子を送信、記録

通知サーバ

* 管理者のアプリ画面　GPSトラッカー（仮アプリ名）　 （仮アイコン）



* 濃厚接触情報、陽性情報はローカルに保存する。
* デバイスとのペアリング

セントラル：スマートフォン、ペリフェラル：デバイス

表示はLocal Nameを使用する。”GPS(仮)”+アルファベット2桁+連番5桁とする。

最初の接続時はデバイスとペアリングする。

次回同じデバイスと接続する時は交換した鍵を保存しておくボンディングを使用する。

Bluetoothペア設定コードはデバイス毎に変わる。

Local Nameの連番5桁の２の補数を計算し、10進数とする。

87654(10)→15666(16)→EA99A(16)→960922(10)

* GPSトラッカーの読み込み処理

BLEのGATT通信でRPI、AEMを読み込む(read)。

UUID 88b9d302-1d53-4743-af14-ccb68179fa75

日時(YYYYMMDDHHMMSS)、生成したTEK(16Byte) + ENIN(4Byte) １４日間分

例

{

{

“time”: 20210115091123,

“TEK”: “a07d635658f82c3c4b8fb211f1e0634”

“ENIN”:[“5fe5bedc”, “5fe5c1ac”, …]←144個

},

繰り返し14個

}

UUID b9428273-c634-491c-9e0a-f3ec17cefbc9

日時(YYYYMMDDHHMMSS)、受信したRPI(16Byte) + AEM(4Byte) １４日間分

例

{

{

“time”: 20210115091123,

“RPI”: “17166ffd32eaf96ce56521975f312572”

“AEM”:[“ff0fa9a9”, “2968d6d0”, …]←144個

},

繰り返し14個

}

* 濃厚接触者の判定

サーバからWeb API（SSLを使用）を使用して情報の送受信を行う。※URL、APIはTBD

データはJSON形式で全データを送り、RPI+AEMは登録済みのキーにて暗号化する。

1. サーバから濃厚接触者のRPI+AEMを取得する。

GET : request body { GPSトラッカーの読み込み処理と同じ形式(json)}

response body { request bodyと同じ形式の濃厚接触者のデータのみ(json) }

1. 濃厚接触情報:0、陽性情報:1＋取得したRPI+AEMと受信したRPI+AEMをマッチングし、一致した場合、濃厚接触とカウントする。TEK+ENINからRPI+AEMの算出はWeb APIで行う。
2. サーバに送信するのは、濃厚接触情報:0、陽性情報:1＋生成したTEK+ENIN１４日間分である。  
   POST : request body { GPSトラッカーの読み込み処理と同じ形式(json) }
   * + アプリを起動していなくても定期的に判定し、アプリのアイコンに通知バッジ（濃厚接触者の合計数）を表示する。

* COCOAからの情報の移行は検討中
* 学校のアプリ画面

管理者のスマートフォンから配信された接触情報を表示する。



※内容はローカルに保存する。

　学校IDとキーを登録時にブローカにユーザ登録する。ユーザ名：学校ID／パスワード：キー

ブローカへのユーザ登録の方法は後日記述

## GPSトラッカー機能

* デバイス（写真は試作品）

電源を入れてから3分間、デバイスとアプリをペアリングできる。  
ペアリング後、アプリで名前、性別、生年月日、キー、位置の送信間隔、有効期限、緊急ボタン押下時間を入力し、BLEを利用してデバイスに保存する。デバイスからの情報はデバイス番号と共に緯度経度、温度、湿度、歩数をLTE-Mを利用して暗号化したMQTT通信でアプリに配信する。中央のボタンを指定時間押下すると、LTE-Mから暗号化しないMQTT通信で近辺のアプリに緊急通知を行い、近辺の登録している方に助けてもらう。



LT-M(MQTT通信)

緯度経度、温度、湿度、歩数

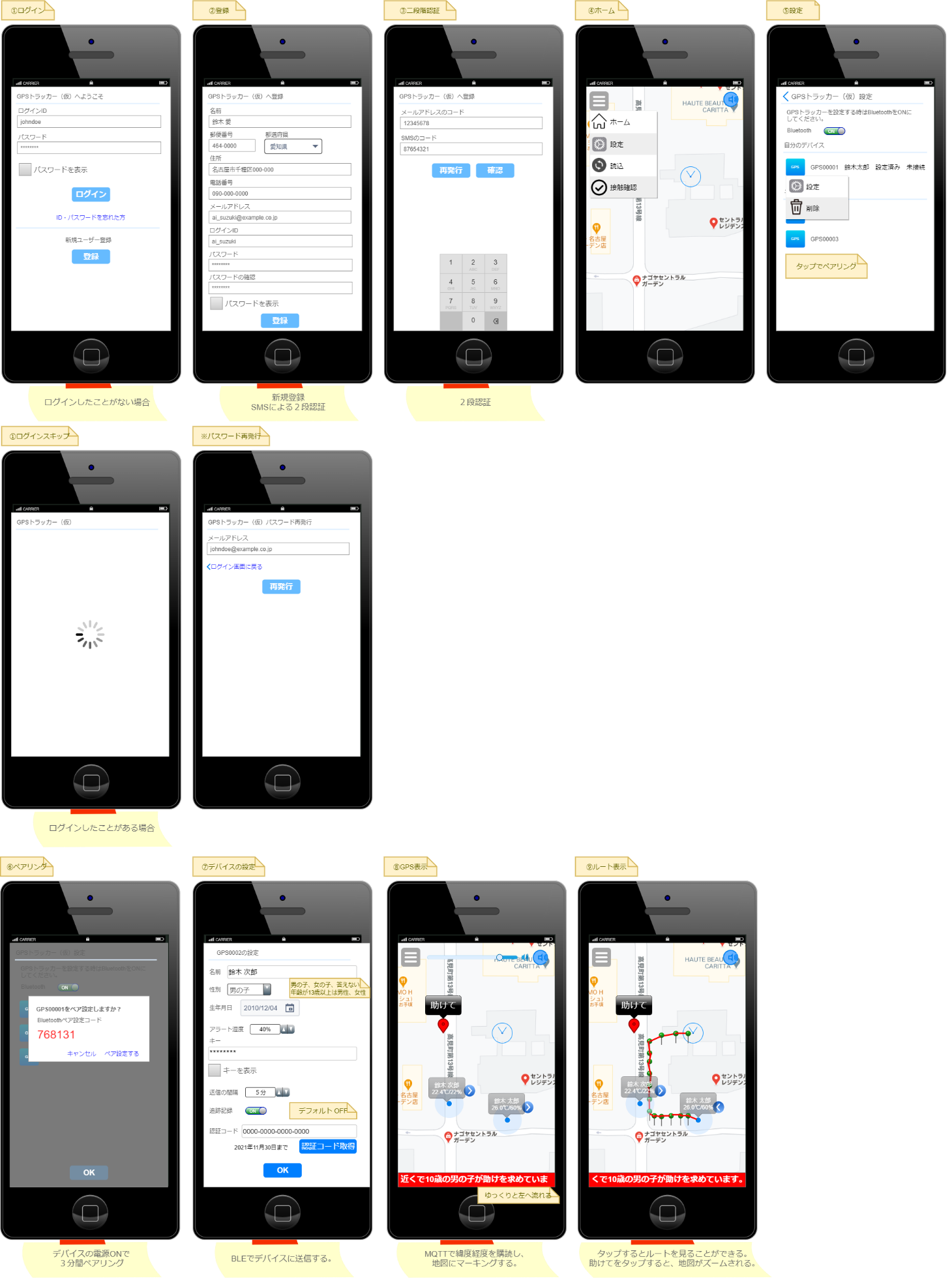


LT-M(MQTT通信)

緊急通知、緯度経度、特徴



* 管理者のアプリ画面



※ログイン情報はサーバ上のプリザンターに保存／読込する。

　スマートフォンは登録時にブローカにユーザ登録する。

　ブローカへのユーザ登録の方法は後日記述

※緊急通知が複数存在する場合は、複数の内容をロールして表示する。

　ルートの２４時間経過したものは削除する。

* GPSトラッカーの設定処理

ペアリングした際にBLEのGATT通信で現在の設定値を読み込み(read)、設定ボタンで設定(write)する。

UUID 51f2e511-be4e-42e2-a502-0bf3aa109855

名前(最大30バイト)、性別(1:男、2:女、0:なし)、生年月日(４バイト)、  
アラート湿度（2バイト）、キー(最大30バイト)、送信の間隔（2バイト）、有効期限(４バイト)

例：

{

“name”:”鈴木太郎”,

“sex”:1,

“alert humidity”:40,

“key”: “wUU,8-cZb(#S$RM”,

“publish interval”:5,

“expiration date”:20211130,

“temporary password”:”e8EPSC-.gJZAMy%”

}

* MQTTの接続

ブローカー・ホスト名：ウィンサーバーを検討中 ポート番号：8883 TLS接続

スマートフォンからの接続は、ユーザ名：ログインID、パスワード：パスワードとする。

デバイスからの接続は、ユーザ名：デバイス名、パスワード：Bluetoothペア設定コードとする。

ただし、登録から翌日０時までの接続はユーザ名：”temporary”、パスワードは登録した時のWeb APIのレスポンス値を使用して接続する。

* MQTTの暗号化

以下の内容でデバイスまたはスマートフォンから暗号化して配信（パブリッシュ）する。購読（サブスクライブ）する側はこのルールで暗号鍵を抽出する。

1. ルール＋暗号鍵をデバイスから配信される。

配信単位：１日毎

topic:デバイス名/”key”/時刻(※1) QoS=2 ※1 YYYYMMDDHHMMSS

メッセージ：

{

{

“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D２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. キーは既にスマートフォンからデバイスにkeyとして設定されているため、ハッシュ値を比較し、その後ろの文字列を暗号キーとして取り出す。

スマートフォンでは、以下のtopicで購読できるようにする。

* + - * デバイス名/#

※　次の暗号鍵を送信する前にそれ以前のメッセージをデバイス側から削除する。

1. 緯度経度がデバイスから配信される。

配信単位：設定時間毎

topic:デバイス名/番号/経過時間(※2) QoS=0 ※2 １．からの秒数

メッセージ：JSON形式 赤字は１．の暗号鍵で暗号化されている。

{  
 “Lat no sec”:秒を切り捨てた緯度,  
 “Lng no sec”:秒を切り捨てた経度,  
 “TEMP”:温度,  
 “HUM”:湿度,  
 “Lat”:緯度,  
 “Lng”:経度,  
 “Step”:歩数,  
 }

スマートフォンでは、以下のtopicで購読できるようにする。（暗号鍵と同じ）

* + - * デバイス名/#

※　暗号鍵を送信する前にそれ以前のメッセージをデバイス側から削除する。

1. 緊急通知がデバイスから配信される。

配信単位：緊急時

常時監視（スマートフォンはアプリが起動していない時も監視する）

topic:デバイス名/emg/10秒単位の緯度/10秒単位の経度 QoS=2

メッセージ：JSON形式

{  
 “Lat”:緯度,  
 “Lng”:経度,  
 “Sex”:0or1or2,  
 “Age”:年齢,  
}

スマートフォンでは、自分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

緊急通知中はデバイス、スマートフォン共にブザーを鳴らす。

スマートフォンに関してはブザーと共にプッシュ通知を行い、プッシュ通知の内容をタップすると、アプリのホームが表示される。

GPSトラッカー（仮）←アプリ名

近くで10歳の男の子が助けを求めています。

　　※　デバイスでボタンの押下がなくなったら削除する。

1. 学校への配信

配信単位：１日単位（スマートフォンから配信）

* 1. ルール＋暗号鍵をデバイスから配信される。

topic:学校ID/デバイス名/”key”/時刻(※1) QoS=2 ※1 YYYYMMDDHHMMSS

メッセージ：”json- Close Contacts”+キー+暗号鍵

キー+暗号鍵は以下のルールで抽出する。

* + 1. 時刻のD２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および学校アプリにkeyとして設定するため、ハッシュ値を比較し、その後ろの文字列を暗号キーとして取り出す。
    - 次の暗号鍵を送信する前にそれ以前のメッセージを管理者アプリ側から削除する。
  1. 緯度経度がデバイスから配信される。

topic:学校ID/デバイス名/時刻(※3) QoS=1 ※3 YYYYMMDDHHMMSS

メッセージ：JSON形式 赤字は画面で設定した暗号鍵で暗号化されている。

{  
 “Close Contacts”:濃厚接触者の数,  
}

学校のアプリでは、以下のtopicで購読できるようにする。

* + - * 学校ID/#

※　送信する前にそれ以前のメッセージをスマートフォン側から削除する。

# 基本シーケンス

基本のシーケンスを記述する。

## 基本機能

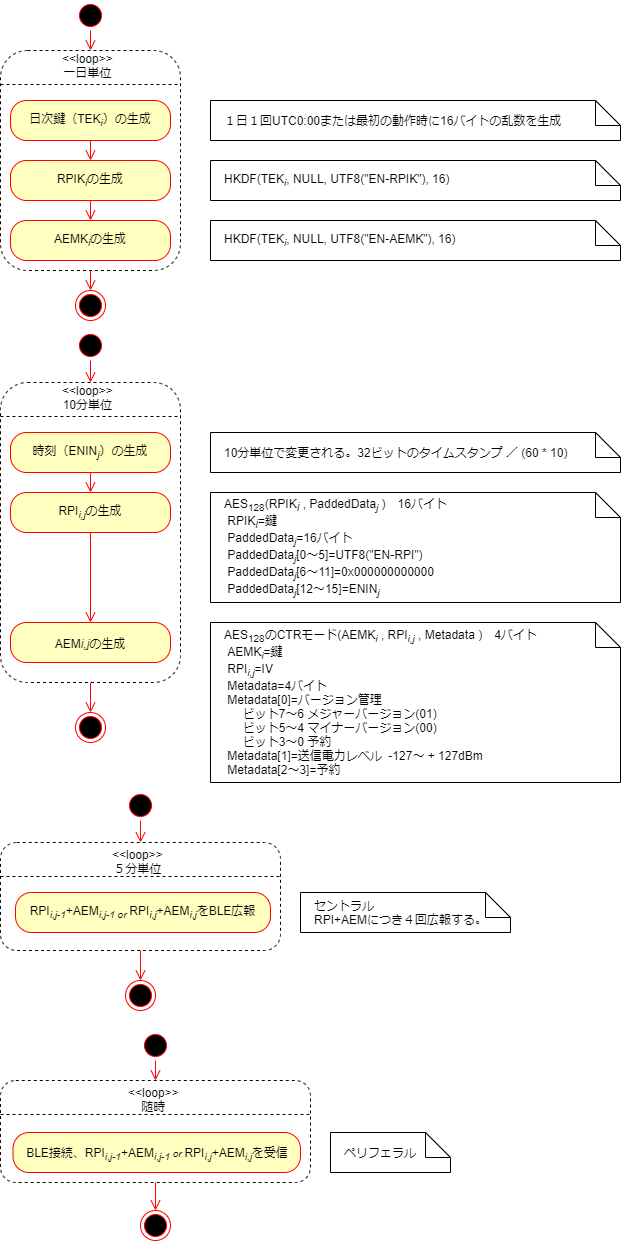
アプリおよびデバイスを動作するにあたっての基本機能のプロセスの流れを示す。



デバイスはアプリと接続し、有効期限を含んだ設定を書き込まれ、有効であれば接触確認機能およびGPSトラッカー機能を動作させる。３分経っても接続しなかった場合、有効期限が有効でない場合は、終了する。

## 接触確認機能

接触確認機能が起動してからのプロセスの流れを示す。



タイムスタンプはUTCを使用

セキュアな乱数を使用

３回以上受信した場合、RPIとAEMを保存する。

スマートフォン、デバイス共常時動作するようにする。スマートフォンは定期的に濃厚接触者の判定も行い、通知バッジ表示を行う。

生成したTEK、ENIN、RPI、AEM、受信したRPI、AEMを14日間保存する。14日間を超えた場合、削除する。

15分以上接触しているかはRPI+AEM※が３回以上一致していることと信号の強さ(RSSI>=-55)で判定する。

* + - RPI+AEMは10分間隔で変更されるが、BLEの送受信は平均１５分となっているため、内部でRPI+AEMが変更されても５分間隔で同じデバイスでスキャンする。

## GPSトラッカー機能

GPSトラッカーが動作開始してからのプロセスの流れを示す。

