

10. 自己診断機能

10.1. 自己診断機能

10.1.1. 自己診断機能概要

BT-2000 では内蔵デバイスの状態確認を目的として、自己診断機能を搭載しています。自己診断機能で診断可能なデバイスと内容は下記の通りです。

表 10-1 自己診断機能での診断対象

デバイス種類	表示名	診断方法
CPU	CPU	エラーチェック
電源制御 IC	PMIC	電源状態確認
メインメモリ (RAM)	DDR	RAM データの確認
データメモリ (Flash)	eMMC	データ領域への読み書き
無線 LAN モジュール	WLAN	デバイスとの通信診断
Bluetooth 3.0	Bluetooth	デバイスとの通信診断
Bluetooth Low Energy	BLE	デバイスとの通信診断
オーディオ IC	Audio IC	デバイス情報固有の読出し
音声コマンド IC	Voice Command	デバイス情報固有の読出し
センサ	Sensor MCU	通信チェック
IMU センサ	IMU	センサデータの取得
照度センサ	Sensor Illumination	センサデータの取得
充電 IC A	Charger IC A	ステータスの確認
充電 IC B	Charger IC B	ステータスの確認
バッテリーA	Battery A	ステータスの確認
バッテリーB	Battery B	ステータスの確認

なお、自己診断機能はあくまで簡易診断です。デバイス内の個別機能の不良等は発見できない場合があります。また、一部の機能は未実装です。

10.1.2. 自己診断機能の利用方法

自己診断機能はアプリとして提供されています。アプリ一覧から“SelfDiagnosis”を起動してください。

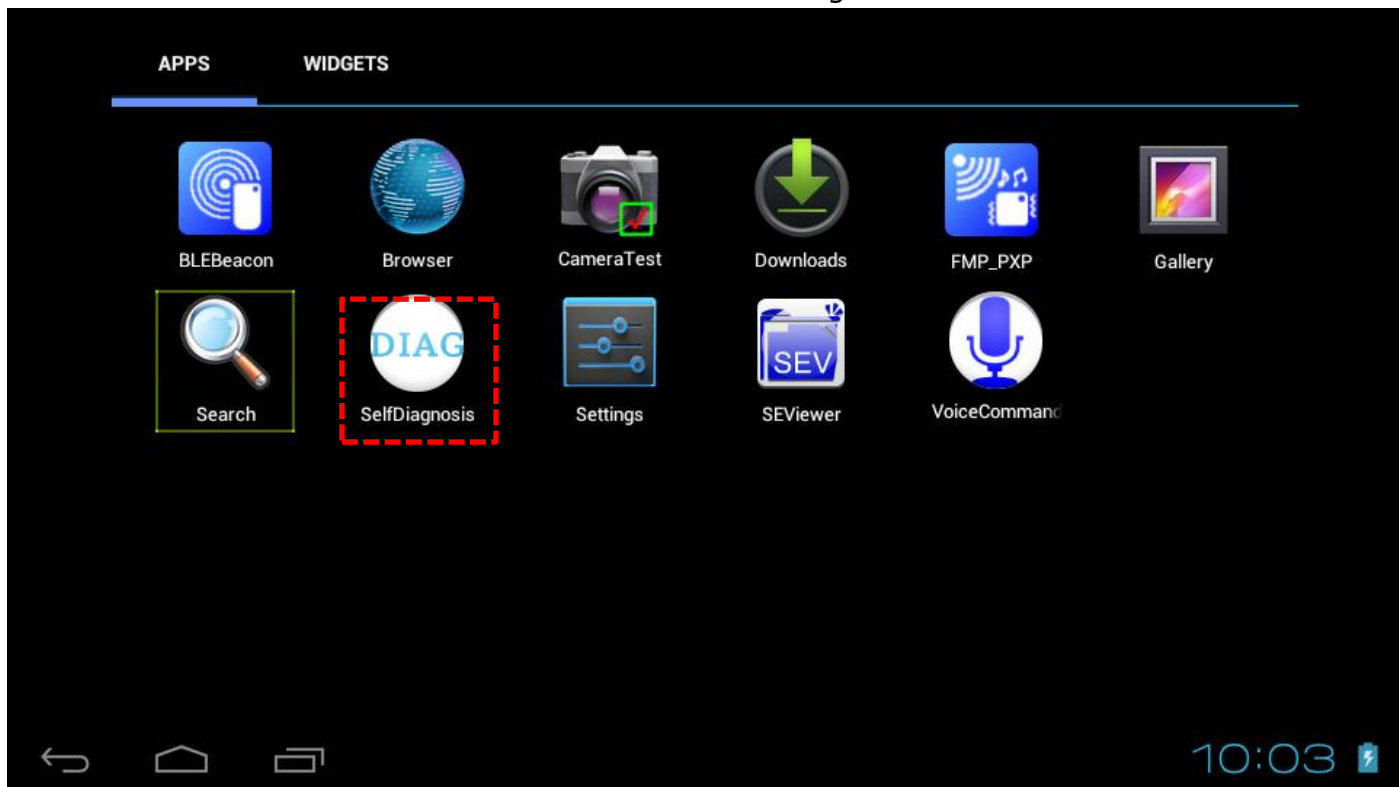


図 10-1 SelfDiagnosis アプリの起動

アプリが起動すると、画面上部に“Start”ボタンがありますので、これを押して診断を開始してください。



図 10-2 SelfDiagnosis アプリの起動画面

診断が実行されると、各デバイスの診断結果が表示されます。各診断の対象と方法については、前述の表 10-1 をご覧ください。



図 10-3 SelfDiagnosis 診断結果

11. GPSアシスト

11.1. GPS アシスト機能概要

BT-2000 は、GPS 測位時間が長くなるコールドスタート時の GPS 初回測位時間（Time To First Fix : TTFF）を短縮する機能として、GPS アシスト機能を搭載しています。あらかじめ GPS 衛星軌道予測データ（EPO データ）をダウンロードし、BT-2000 に書き込んでおくことで、衛星データのダウンロード時間を短縮し、その結果 TTFF を短縮することができます。

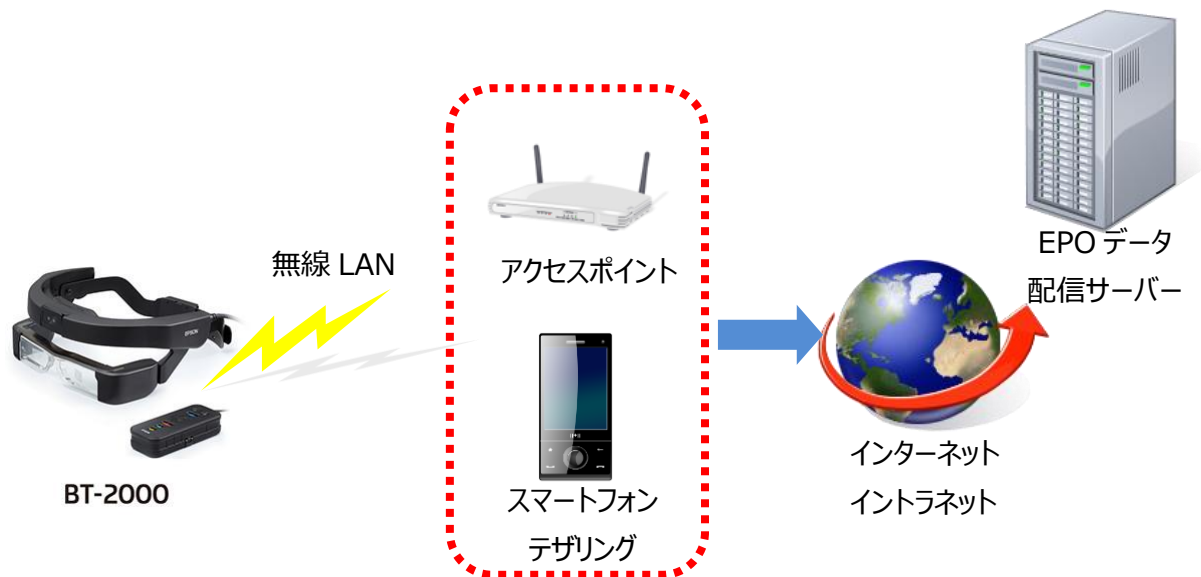


図 11-1 EPO データダウンロードイメージ

11.2. GPS アシスト機能使用・開発時の注意事項・制限事項

■ GPS アシスト機能の効果について

GPS アシスト機能の効果を十分に発揮するには、GPS 衛星を 4 つ捕捉する必要があります。

GPS 衛星を捕捉できないもしくは、捕捉しにくい環境下では十分に効果を発揮できない場合があります。

■ GPS のコールドスタートについて

GPS の測位を開始する方法には、前回測位した衛星情報を利用するホットスタート、およびウォームスタート、前回の測位情報を利用できない場合のコールドスタートがあります。

GPS アシスト機能はコールドスタート時に効果を発揮する機能です。ホットスタート、ウォームスタート時の TTFF 短縮には効果を発揮しません。

■ 時刻設定について

GPS の衛星位置を予測するために時刻情報を利用します。そのため、システムの時間を正確に合わせてください。

■ EPO データのダウンロードについて

EPO データの有効期間は 30 日ですが、新しいデータを利用する方が GPS アシスト機能の効果が高いため、1 週間を目安に EPO データをダウンロードしなおしてください。

■ EPO データ配信サーバーについて

BT-2000 は EPO データをインターネット経由でエプソンの EPO データ配信サーバーよりダウンロードすることができます。

インターネットを利用できない、ローカルイントラネットで GPS アシスト機能を利用する場合は、エプソンの EPO データ配信サーバーより、EPO データをコピーし、ローカルイントラネットへ配置してください。

BT-2000 に EPO データ配信サーバーを設定するには、“内部ストレージ¥agps¥agps.conf”に配信サーバーの URL を記載してください。

■ GPS アシスト機能 API について

BT-2000 の GPS アシスト機能 API はエプソン独自 API を採用しております。

そのため GPS アシスト機能 API を使用するアプリケーションを開発する場合、エプソンが提供する SDK を組み込み、エプソン独自 API をご使用ください。

11.3. 使用方法

GPS アシスト機能の使用方法を記述します。

11.3.1. EPO データのダウンロード



- Wi-Fi 機能を ON にし、EPO 配信サーバーへアクセスできるようにしてください。
エプソンの EPO データ配信サーバーへアクセスするにはインターネット環境が必要です。
イントラネットへ EPO データを配置した場合は、イントラネットへ接続が必要です。
- [設定]アプリを開き、[位置情報サービス]を選択します。



- [GPS 機能]を ON にして[AGPS データアップデート]を選択します



11.3.2. EPO データの書込み

EPO データを BT-2000 へ書込んでいる間は、システムバーに  のアイコンが表示されます。
 のアイコンが消えれば書込みは完了です。

11.3.3. EPO データの利用

- GPS を利用するアプリを起動してください。コールドスタート時に GPS アシスト機能が自動的に動作します

11.4. 設定方法

EPO データの書き込みに関する設定について記述します。

設定ファイルは“内部ストレージ¥agps¥agps.conf”にあります。

通常は変更不要ですが、EPO データをローカルのサーバーに配置したい場合など、必要に応じてファイルを編集してください。

11.4.1. 設定ファイル agps.conf

ファイルパス： /mnt/sdcard/agps/agps.conf

デフォルト： 無し。[設定]-[AGPS データアップデート]が実行されたときに既定値で生成されます。

ファイルフォーマット： java.util.Properties のフォーマットに従って、記載してください。

URL 表記で使用するコロンには、“¥:”を用いています。

表 11-1 agps.conf の設定項目

No	設定名	説明
1	EPO_URL	EPOデータ配信サーバーのURLを設定してください。 デフォルト値は“ https¥://tech.moverio.epson.com/epo/data.bin”
2	EPO_NUM_SEGS	EPO書き込みセグメント数（1～28）を設定してください。 1セグメント=6時間となります。 デフォルト値は28(6時間×28=7日)
3	LAST_MOD	前回のダウンロード時刻が表示されます。 設定は不要です。

11.5. アプリケーションインターフェース機能概要

EPO データのダウンロード、および BT-2000 への書き込みに関する API について記述します。

- 1) EPO データを EPO データ配信サーバーからダウンロードする
- 2) ダウンロードした EPO データを、BT-2000 の GPS モジュールへ書き込みます。

なお、各 API を使用するためには、下記モジュールをインポートしてください。

```
com.epson.moverio.bt2000.agpsctl
```

表 11-2 AGPSControl クラスのインターフェース一覧

NO.	関数名	機能概要	備考
1	downloadEpo	設定ファイルagps.confに記載された設定でEPOデータをダウンロードする	EPSON独自API
2	downloadEpo_url	引数で指定したURLからEPOデータをダウンロードする	EPSON独自API
3	writeEpo	設定ファイルagps.confに記載された設定でEPOデータをBT-2000へ書き込む	EPSON独自API
4	writeEpo_segs	引数で指定したセグメント分のEPOデータをBT-2000へ書き込む	EPSON独自API

11.6. アプリケーションインターフェース詳細

11.6.1. downloadEpo

- 機能

設定ファイル(/sdcatd/agps.conf)に記載された設定で EPO データをダウンロードします。

EPO データはエプソンの EPO データ配信サーバーより配信されます (毎日 0:10(UTC)更新)

- 書式

```
int downloadEpo(void);
```

- パラメータ

なし

- 戻り値

常に 0

- ブロードキャストインテント

Intent	ステータス
AGPS_DOWNLOAD_COMPLETED	ダウンロード完了
AGPS_DOWNLOAD_ERROR	ダウンロードエラー
AGPS_CONF_ERROR	agps.confエラー (agps.confがない、もしくはEPO_URLに不備がある)
AGPS_NETWORK_ERROR	Wi-Fi OFFによるエラー
AGPS_FILE_NOT_UPDATED	サーバー上のファイルが更新されていない

11.6.2. downloadEpo_url

- 機能

引数に指定した URL から、EPO データをダウンロードします。

- 書式

```
int downloadEpo_url(String arg0);
```

- パラメータ

型 名前	説明
String arg0	EPOデータ配信サーバーのURL

- 戻り値

常に 0

- ブロードキャストインテント

downloadEpo と共通。

11.6.3.writeEpo

- 機能

agps.conf に設定されたセグメント数分の EPO データを、BT-2000 の GPS モジュールへ書き込みます。

- 書式

```
int writeEpo();
```

- パラメータ

なし

- 戻り値

常に 0

- ブロードキャストインテント

Intent	ステータス
AGPS_WRITE_COMPLETED	書き込み完了
AGPS_WRITE_ERROR	書き込みエラー
AGPS_CONF_ERROR	agps.conf エラー（agps.conf が ない、もしくは EPO_NUM_DAYS に不備がある）
AGPS_FILE_EXPIRED	EPO ファイルが古い（現時点で期限切れ、指定期間分のデータが無い）

11.6.4.writeEpo_segs

- 機能

引数で指定したセグメント数分の EPO データを、BT-2000 の GPS モジュールへ書き込みます。

- 書式

```
int writeEpo_segs(int arg0);
```

- パラメータ

型 名前	説明
int arg0	書き込むセグメント数

- 戻り値

常に 0

- ブロードキャストインテント

WriteEpo と共通。

11.7. GPS アシスト用 API の利用方法

11.7.1. EPO データのダウンロード

AGPSControl のインスタンスを生成し、EPO データをダウンロードします。

■ ダウンロード時の実装例

```
public void execDownload(View view) {  
    Log.d(TAG, "exec Download");  
  
    try {  
        AGPSControl ctl = new AGPSControl();  
        ctl.downloadEpo();  
    } catch (Exception ex) {  
        ex.printStackTrace();  
    }  
}
```

11.7.2. インテントの受信方法

ダウンロードや書き込みの処理は非同期で行なわれるので、直接戻り値を参照できません。

そのため、ブロードキャストインテント(BroadcastIntent)を使い、処理の結果を取得してください。

各 API が発行するブロードキャストインテントの値については、各 API のブロードキャストインテントの欄を参照ください。

ここでは、インテントの受信方法について説明します。

11.7.2.1. 起動時

アプリケーション起動時にインテントフィルタとレシーバを設定します。

■ 実装例

```
private MyReceiver receiver = new MyReceiver();
private IntentFilter filter = new IntentFilter();

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    filter.addAction(AGPSControl.AGPS_CONF_ERROR);
    filter.addAction(AGPSControl.AGPS_DOWNLOAD_COMPLETED);
    filter.addAction(AGPSControl.AGPS_DOWNLOAD_ERROR);
    filter.addAction(AGPSControl.AGPS_FILE_EXPIRED);
    filter.addAction(AGPSControl.AGPS_FILE_NOT_UPDATED);
    filter.addAction(AGPSControl.AGPS_NETWORK_ERROR);
    filter.addAction(AGPSControl.AGPS_WRITE_COMPLETED);
    filter.addAction(AGPSControl.AGPS_WRITE_ERROR);

    getApplicationContext().registerReceiver(receiver, filter);
}
```

■ インテントレシーバの例

```
class MyReceiver extends BroadcastReceiver {
    public MyReceiver() {
        super();
    }
    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equals(AGPSControl.AGPS_CONF_ERROR)) {
            Toast.makeText(getApplicationContext(), "AGPS_CONF_ERROR",
                Toast.LENGTH_LONG).show();
            Log.d(TAG, "Intent Received | AGPS Conf Error");
        }
        if (intent.getAction().equals(AGPSControl.AGPS_DOWNLOAD_COMPLETED)) {
            Toast.makeText(getApplicationContext(), "AGPS_DOWNLOAD_COMPLETED",
                Toast.LENGTH_LONG).show();
            Log.d(TAG, "Intent Received | AGPS Download Completed");
        }
        ~~~~~(他のインテントについては省略)~~~~~
    }
}
```

11.7.2.2. アプリケーションの状態遷移に伴うレシーバの再設定

アプリケーションがバックグラウンドになるなどの状態遷移をしたときには、インテントレシーバを解除してください。

■ レシーバ解除の例

```
@Override
protected void onPause() {
    super.onPause();
    getApplicationContext().unregisterReceiver(receiver);
}
```

再びアプリケーションに戻った場合には、再度インテントレシーバを登録してください。

■ 再登録の例

```
@Override
protected void onResume() {
    super.onResume();
    getApplicationContext().registerReceiver(receiver, filter);
}
```

— Memo —