

組込み用途における「Acty-G3」アプリ開発の実例

CYBERDYNE Omni Networks

Acty-G3とは？

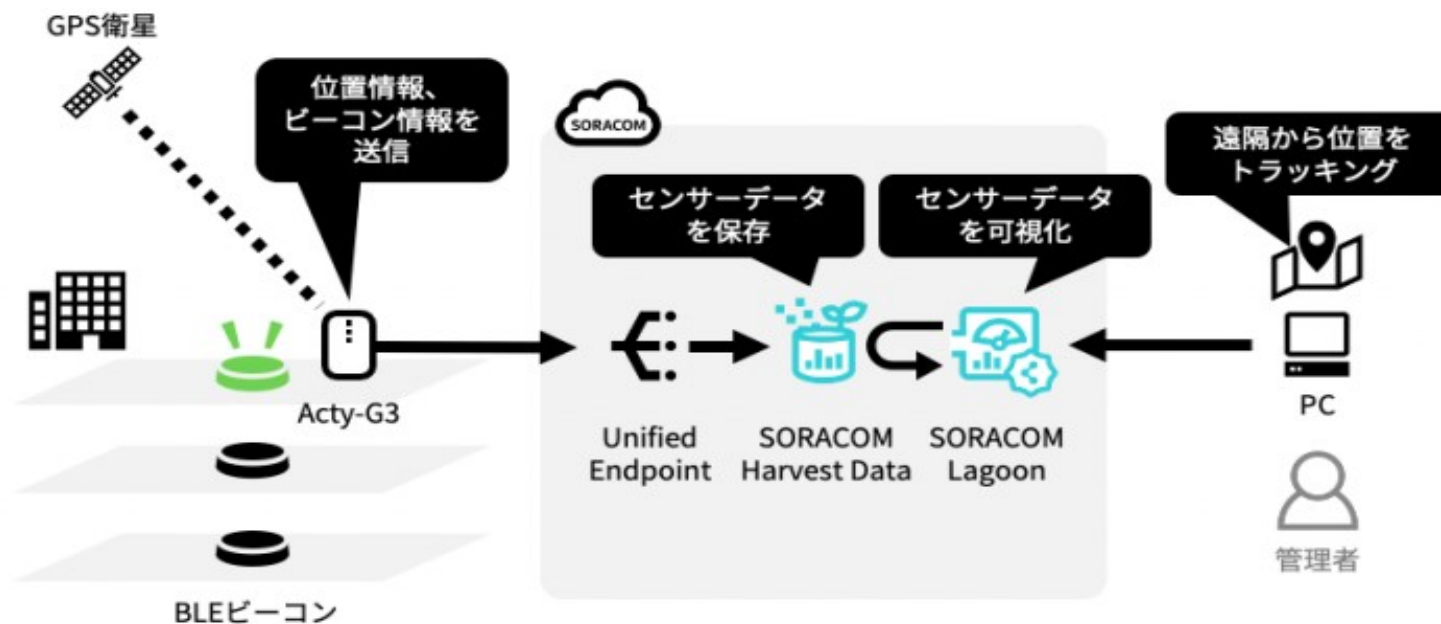
- Android10 Goを採用。
- Andorid用のアプリが動作する。
- 画面や操作デバイスがない。



開発環境

- Androidであるため専用のSDKはない。
- GoogleからAndroid Studioをダウンロードすれば開発を始められる。
- デベロッパー向け公式サイト Android Developers
- 個人・法人を問わず多くのアプリの開発が行われており、情報の収集が容易。
- XDA DevelopersやStack Overflowといったコミュニティサイトも有用。

SORACOM Harvestアプリ概要



GPS Tracker connected with SORACOM

- 位置情報の取得。
- 取得した情報をサーバに送信。
- 電源オン後、アプリは自動的に起動。
- ただし開始と停止はボタンで操作。
- 動作状態をLEDで表示。

位置情報を取得して送信する

- GPS信号はAndroidが受信しているので、アプリは定期的にそれを受け取るだけ。
- 位置情報はGPS信号のほかに、Wi-Fi、基地局といった情報も加味する必要がある。
- FusedLocationProviderを使用すれば、状況に応じてそれらをアプリ側で調整する必要はない。
- データはJSON形式に整えてhttp clientで送信。

ボタンの検出

ブロードキャストレシーバーを立ててIntentを受け取る

```
private final BroadcastReceiver mIntentReceiver = new BroadcastReceiver() {  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        Log.d(TAG, "onReceive intent=" + intent.getAction());  
        if(intent.getAction().equals("covia.intent.action.BUTTON1_SHORT_PRESS")){  
            Log.d(TAG, "Receive covia.intent.action.BUTTON1_SHORT_PRESS !!!");  
            if (!isWorking) {  
                startTesting();  
            } else {  
                stopTesting();  
            }  
        }  
    }  
};
```

LEDの制御

制御ファイルに書き込み

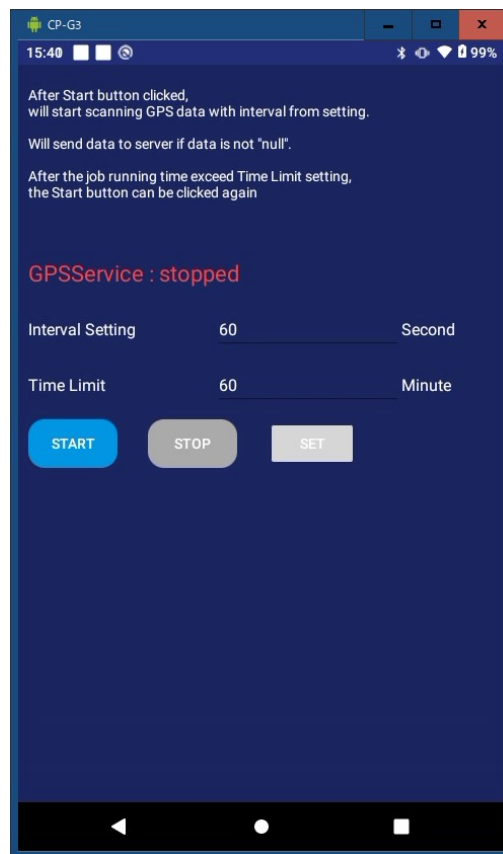
```
public void Led1_RedOn() {  
    try {  
  
        File file=new File("/sys/class/leds/led1-red/brightness");  
  
        FileOutputStream fos=new FileOutputStream(file) ;  
  
        fos.write(String.valueOf("255").getBytes());  
  
        fos.close();  
  
    } catch (IOException e) {  
  
        e.printStackTrace();  
  
    }  
}
```


アプリの動作まとめ

- BOOT_COMPLETEを受けてアプリは自動起動。
- ボタンによって動作の開始／停止。
- LEDで動作状態を表示。
- FusedLocationProviderで位置情報を取得。
- 得られたデータをhttp clientで送信。

ファンクションボタンとLEDの制御だけがG3固有の実装

位置情報を取得して送信する



- 最低限のUIは用意。
- 送信周期と送信期間の設定。
- 操作はPCからVysorやscrcpyなどの画面転送ソフトを利用。

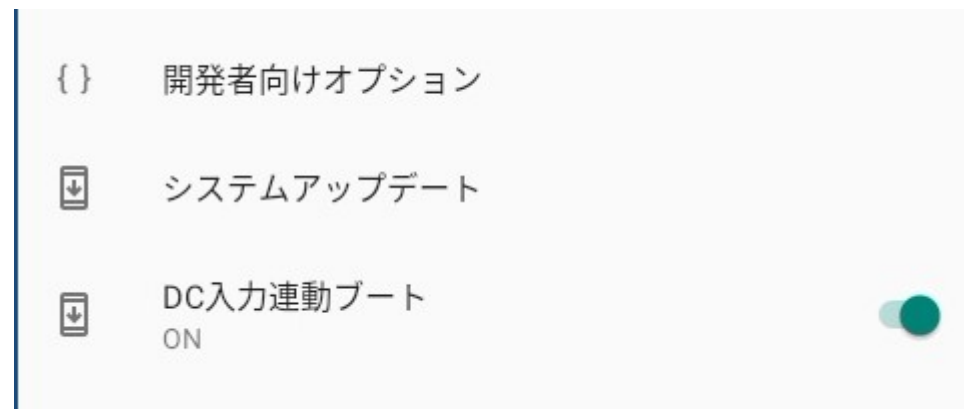
車載GPSトラッカーの例

- エンジンスタートと連動して起動させたい。
- エンジンストップ後はシャットダウンさせたい。
- つまり、ドライバーの操作を不要にしたい。

このためには、電源周りの制御が必要になる。

- 外部電源供給による自動的な電源オン。
- 任意のタイミングでのシャットダウン／再起動。

DC入力連動ブート



- ファームウェアレベルでサポート。
- 外部電源が供給されるとG3を自動的に起動させる。

シャットダウンさせるには？

エンジンが停止すると？

→ バッテリー動作に切り替わる。

→ アプリから充電状態を検出して、Dischargingになったらシャットダウンすればよい。

しかし、問題が . . .

ユーザーアプリはシャットダウンを実行できない。

Permission (権限)

- Normal , Dangerous , Signatureの3段階。
- アプリは機能に応じたPermissionを宣言する必要がある。
- ネットワークを使うには INTERNETを宣言 = Normal Permission
`uses-permission android:name="android.permission.INTERNET"`
- GPSを使うにはACCESS_FINE_LOCATIONを宣言 = Dangerous Permission
`uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"`
- ユーザーアプリから使えるのは、NormalとDangerousまで。

Signature Permission

- Signature権限を使うにはPlatform Keyが必要。
- 通常はPlayストアなどのGoogleアプリや端末メーカーがプリインストールで提供するアプリにしか許可されない。
- 電源周りの制御にはAndroidManifestで特権ユーザーの宣言する。

```
android:sharedUserId="android.uid.system"
```

これには Signature権限が必要になる

Platform Key

- アプリ開発者向けに提供。
- アプリからシャットダウンや再起動が可能になる。
- Signature権限を必要とする他の機能も利用可能。
- 例えばIMEIの取得には、`READ_PRIVILEGED_PHONE_STATE`を宣言するが、これにもSignature権限が必要となる。

Runtime permission



- Android6.0からの仕様。
- Dangerous permissionを要求した場合、アプリ起動時に承認を求める。
- 特権ユーザーになることで回避可能。

アプリのセルフアップデート

- アップデート機能は必須。
- Playストアも利用可能だが、端末1台ごとにはGoogleアカウントの設定が必要となる。
- 運用台数が多いほど現実的ではない。
- 独自にアップデートサーバを用意。
- バックグラウンドインストールを使用すべきだが、これにはシステムと同じ権限が要求される。

スケジュールによる起動

特定の時間内だけ動作させたい

- 店舗や倉庫の営業中だけ。
- バッテリーを節約するため指定された時間内だけ。

AlarmManagerの拡張で対応 ※指定した時間後にG3を起動させる

```
alarmmanager.setExact(7, cal.getTimeInMillis(), pendingintent);
```

この拡張についてのドキュメント、サンプルアプリとソースコードも提供。

補助アプリ

- アプリのインストール／アップデートの補助。

G3_Help_Install_APK

- シャットダウン、再起動、指定時間後の起動を補助。

G3_Help_Shutdown

アプリからintentを送ることによって、それぞれの機能呼び出す。