

# 港湾空港技術研究所 資料

TECHNICAL NOTE

OF

THE PORT AND AIRPORT RESEARCH INSTITUTE

**No.1312**      September 2015

流出油のリアルタイム捕捉に向けた簡易漂流テレメトリブイの開発と応用

藤田 勇  
松崎 義孝

国立研究開発法人 港湾空港技術研究所

National Research and Development Agency,  
Port and Airport Research Institute, Japan

## 目 次

要 旨 .....	3
1. まえがき .....	4
2. 流出油捕捉用簡易漂流テレメトリブイの開発 .....	4
2.1 開発の背景 .....	4
2.2 漂流ブイの概要と機械要素 .....	5
2.3 電子回路構成 .....	6
2.4 ソフトウェア構成 .....	8
3. 簡易漂流テレメトリブイの実海域試験と流出油捕捉 .....	13
3.1 機能試験 .....	13
3.2 漂流油拡散試験 .....	17
3.3 漂流油のリアルタイム捕捉の検討 .....	19
4. その他の応用 .....	20
4.1 漂流物捕捉ツールへの活用 .....	20
4.2 気球空撮システムへの応用 .....	21
5. まとめ .....	23
謝辞 .....	23
参考文献 .....	23

# **Development of Smartphone-embedded Telemetry Drifting Buoy for Tracking Spilt Oil and its Applications**

**Isamu FUJITA\***  
**Yoshitka MATSUZAKI\*\***

## Synopsis

We developed a new drifting buoy using Android smart phone. The drifting buoy can detect its position using GPS system and also measure wind velocity and direction. The measured data can be immediately transmitted to a distant location by 3rd or 4th generation mobile phone network. Field tests were conducted to verify its basic function and found that the system can work in nearshore area for many hours without human intervention nor network communication errors. The developed buoys were applied to field tests for measuring horizontal turbulent diffusivity of floating thin oil and tracking drifting oil patch position. In addition to oil spill response applications, other potential applications using the Android smart phone as an embedded system are also proposed.

### **Key Words:**

Android smart phone, telemetry, drifting buoy, spilled oil, turbulent diffusivity

---

\* Head, Oil Spill Response Research Group, Frontier Technology R&D Field  
\*\* Researcher, Multiphase Flow Group, Coastal and Estuarine Environment Field  
3-1-1 Nagase, Yokosuka, 239-0826 Japan  
Phone : +81-46-844-5065 Fax : +81-46-844-0575 e-mail: fujita@pari.go.jp

# 流出油のリアルタイム捕捉に向けた 簡易漂流テレメトリブイの開発と応用

藤田 勇\*  
松崎 義孝\*\*

## 要 旨

海上漂流油の捕捉技術として、スマートフォンを活用した海象情報のリアルタイム収集用の漂流ブイを開発した。開発したブイでは、GPSによる測位と風向風速を計測し、3Gあるいは4Gの携帯電話網を介してリアルタイムにデータを遠隔地に伝送することができる。実海域において機能試験を実施し、ブイが正常に機能すること、長時間無人で運用できること、沿岸部の使用で通信に問題がないこと等を確認した。開発したブイは実海域において、流出油の水平乱流拡散係数を計測する実験並びに、漂流油の位置の捕捉実験に活用した。漂流ブイが、実験用機材としても漂流油の捕捉用機材としても有効なツールであることを示した。更に要素技術的な応用として、アンドロイドスマートフォンを組み込み用途に用いた他の例を紹介した。

**キーワード：**アンドロイドスマートフォン，テレメトリ，漂流ブイ，漂流油，乱流拡散

---

\*新技術研究開発領域 油濁対策研究チームリーダー  
\*\* 新技術研究開発領域 油濁対策研究チーム（併任） 主任研究官  
〒239-0826 横須賀市長瀬3-1-1 港湾空港技術研究所  
電話：046-844-6065 Fax：046-844-0575 e-mail:fujita@pari.go.jp

## 1. まえがき

海上における浮遊物の漂流を把握かつ予測することは、洋上での回収をはじめとして漂流物への対応を考える上で重要である。海上での浮遊物としては、浮遊ゴミや船舶の事故等によって発生する流出油などが該当する。著者らはこれまで、海上流出油を主な対象として、油拡散粒子モデルによる漂流予測シミュレーションの開発に取り組んで来た<sup>1)</sup>。油をはじめとして海上浮遊物の漂流を予測する上では、海流、潮流、吹送流といった海水の運動による漂流と洋上風から受ける影響の両者の再現が求められる。著者らが開発した漂流予測シミュレーションでは、海流と風に関して、気象庁が発行しているGPV予報値を使用し、潮流に関しては、調和解析により計算される値を用い、拡散係数など油の拡がり方に関するパラメータを考慮して計算している。今後の予測精度向上に向けての開発課題としては、実際に洋上で計測された実測データとシミュレーションとのデータ同化などが考えられる。一つには、HFレーダーを用いた表層流のリモートセンシング技術の活用が考えられ、これは漂流ゴミの回収作業などの実務にも活用されるようになってきているが<sup>2)</sup>、洋上風の影響を敏感に受ける浮遊油に対してどのように適用していくかは、今後の検討を待つ必要がある。一方、洋上において直接計測されるデータ(Sea truth)の活用も重要で、油の漂流位置を把握する目的に対して、漂流ブイは有効な手段の一つと考えられており、近年では油を自動で追跡するブイ<sup>3)</sup>などの開発が進められるなど、これまでも色々なものが提案されている。しかしながら、Fingasのレビュー<sup>4)</sup>によると、これまで開発されたブイについては、少数の例外を除いて、ほとんどのパッシブブイが漂流油への追従性が悪く、更に観測機能や通信機能を有する高機能ブイでは、全ての機種において油に追従しなかったことを報告し、その原因を風の影響であるとしている。計測や通信機器を内蔵するブイでは、どうしてもある程度の体積を要するため、ブイに油と同じ漂流特性を与えることは難しい。しかし、ブイ自体が完全に油に追従して流れない場合でも、ブイの位置における風向風速がわかれば、その影響を考慮することで、ブイの位置を基準として油の位置を推定することが原理的には可能である。本研究では、このような着想に基づいて、漂流ブイに風向風速の計測機能を付加して、位置情報とともに無線通信でリアルタイムに情報を伝送できるブイを試作検討した。

究極的な漂流ブイとしては世界中のどこでも使用できて且つ安価であるというのが理想であるが、両者を同時に実現することは難しいことから、本研究では、使用できる海域を沿岸域に限定することで、簡易且つ安価なシステム構

築を目指した。具体的には携帯電話のポケット通信を活用することを考え、ブイに内蔵する電子回路としてアンドロイドOSを搭載したスマートフォンをそのまま使い、スマートフォンの内部機能を活用することで難しい電子回路設計などを出来る限りなくし、機能実現のためのプログラミングもアンドロイド用のライブラリを活用するなどして、開発負荷が小さく且つ安価なシステムを構築した。

本資料では、2章においてブイの開発過程を紹介する。機械要素、電子回路構成や、制御プログラムの詳細を述べる。スマートフォンを活用した他の応用を検討する読者の便を考え、部分的ながらプログラムリストも加えることで、開発の具体的なイメージを捉えられる様に配慮した。主として応用に関心のある読者は、2-4のソフトウェア構成は飛ばしていただいて差し支えない。

3章においては、開発したブイを用いた実海域試験について述べている。ブイの機能試験結果に続いて、漂流油の乱流拡散係数を求める為の拡散実験を紹介し、モデル化に向けた考察を行う。更にブイを用いた漂流油のリアルタイム捕捉の方法を検討し、開発した漂流ブイの有用性を検証する。

更に4章において組み込み機器としてスマートフォンを活用するその他の応用例として大型漂流物の捕捉への応用と、気球空撮システムのペイロードとしての応用例を紹介する。

本研究の動機と主題は漂流油の捕捉技術であるが、伏線としては、伸長著しいICT技術やIoT (Internet of Things)などの活用があり、こうした視点においても読者の興味を引くことができれば幸いである。

## 2. 流出油捕捉用簡易漂流テレメトリブイの開発

### 2.1 開発の背景

近年携帯電話の普及は目覚ましいものがある。なかでもスマートフォンと呼ばれる高機能携帯電話は、さながら広域デジタル通信機能を内蔵した小型のコンピュータの様相を呈している。従来の携帯電話との違いは、性能の向上に加えて目的に応じた諸々のアプリケーションを比較的簡単に開発して実装することができるようになってきている点である。例えばアンドロイドOSを搭載したスマートフォンではAPI(Application Programming Interface)の仕様が一般に公開されており、ユーザは、それら呼び出すプログラミングをJavaにより記述することでスマートフォンの機能をカスタマイズして利用することができる仕組みになっている。従来組み込み機器の開発は多大な労力を要するものであつ

たが、スマートフォンを活用することで比較的容易に必要な機能を実現することができる。開発環境に関しても、Eclipseといった無料の統合開発環境が提供されており、これにアンドロイド関連のプラグインを追加することで、容易に構築することができる。また開発関連の情報も書籍やインターネット上に多く提供されており、初めて開発に着手する技術者にとっても、比較的敷居が低いものとなっている。

著者らはこうした利点に着目し、流出油の捕捉技術、ならびに実海域試験での計測ツールに活用することを考え、比較的安価な材料費で構築可能な簡易漂流ブイによる遠隔テレメトリ技術に取り組むこととした。将来的な発展としては図-1に示すように海上流出油のリアルタイムモニタリングシステムにおけるデータ同化用のリアルタイムデータ計測システムとして漂流シミュレータおよび操船支援システムの一部に組み入れていくことを考えているが、本資料においては、漂流ブイのプロトタイプの開発内容について紹介する。

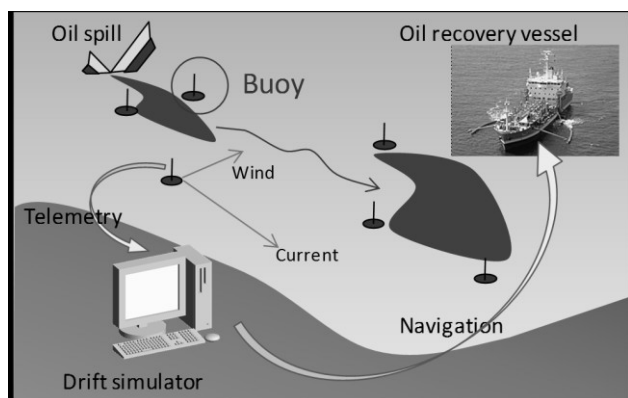


図-1 漂流ブイを活用した漂流油捕捉イメージ  
Oil spill: 油流出, Buoy: ブイ, Telemetry: テレメトリ, Drift simulator: 漂流シミュレータ, Navigation: 操船情報, Oil recovery vessel: 油回収船, Wind: 風, Current: 流れ

## 2.2 漂流ブイの概要と機械要素

スマートフォンを実装した漂流ブイの開発にあたっての測定項目は、GPSによる測位情報、洋上の風向、風速とし、2012年度に一次試作を実施し<sup>5)</sup>、実海域における動作確認試験を行った。それら一次評価の結果を踏まえて、2013年度に二次試作を実施した<sup>6)</sup>。ここでは二次試作機の構成を紹介する。漂流ブイの主な緒元を表-1に、構造と外観を図-2及び写真-1に示す。

ブイの胴体部は円筒形状とし、材料には塩ビのパイプを使用し、内部にスマートフォンをはじめとする電子機器や

バッテリー類を組み入れた。ブイの浮力調整と動揺抑制のために、胴体周囲に円盤状の浮力体（ポリスチレン製、φ480mm厚さ50mm）を追加した。風速計は胴体部上部に細い塩ビ管のマストを設け、その頭頂部に熱式の風速計を取り付けた。この風速計は単体では風向を計測することができないので、ブイ全体が風により向きを変えることで、風向を検知できるように工夫した。マストに垂直翼を取り付け、風圧によりブイ全体を回転させ、風速計が風に正対するようにした。ブイの向きはスマートフォン内蔵の方位センサーにより検出することとした。ブイの下部には水の流れに対して抵抗となる十字形状の抵抗板を吊り下げられるようにした。ブイと抵抗板の接続はブイの回転を妨げないようにスィベルジョイントを入れた。抵抗板を吊り下げ深度や大きさを変えることで、ブイが追従する水塊の深さや程度を調整することができるようになっている。ブイの本体重量は3.6kgと比較的軽量で、海への投入と回収が一人の作業者で容易にできるものとなっている。

表-1 漂流ブイの主な緒元

Body ( 胴体部 )	
Diameter ( 直径 )	Φ114
Float ( 浮き )	Φ480
Height ( 高さ )	410 ( Main body ) ,280(Mast) 410 ( 胴体部 ) ,280 ( マスト )
Weight ( 質量 )	3.6kg ( battery and drogue excluded ) 3.6kg ( 電池と抵抗板を除く )
Functions ( 機能 )	
Location(測位)	GPS sensor/GPS センサー*
Azimuth ( 方位 )	Magnetic sensor/磁気センサー*
Wind velocity ( 風速 )	Thermal anemometer 0-15m/s 熱式風速計 0-15m/s
Components ( 構成要素 )	
Smartphone ( スマートフォン )	Google Nexus5 (Sim free) Android 4.4
Sim ( シム )	B-mobile 4G Fair 1GB ( Prepaid) B-mobile 4G Fair 1GB ( プリペイド)
I/O Extension ( I/O 拡張 )	Sparkfun IOIO DEV-10748
Anemometer ( 風速計 )	KOA AFS0001
Battery ( 電池 )	48(max.) AA 最大48本, 単三型
*Built-in function of the smartphone *スマートフォンの内部機能	

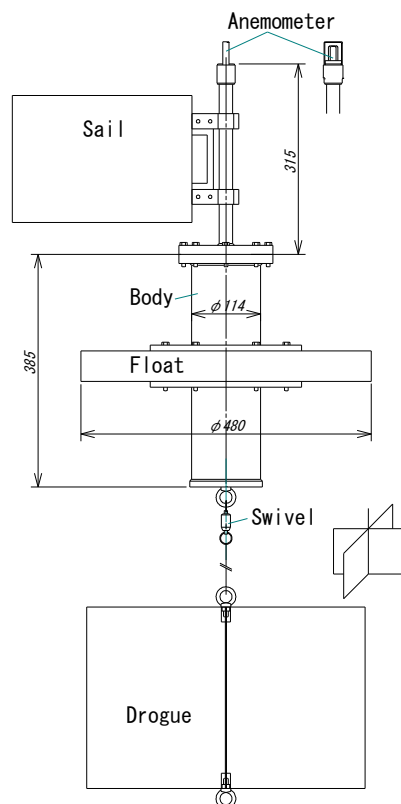


図-2 ブイの機械構成

Anemometer: 風速計, Body: 胴体, Float: 浮き, Drogue: 抵抗板, Swivel: スイベル継手, Sail: 帆(尾翼)

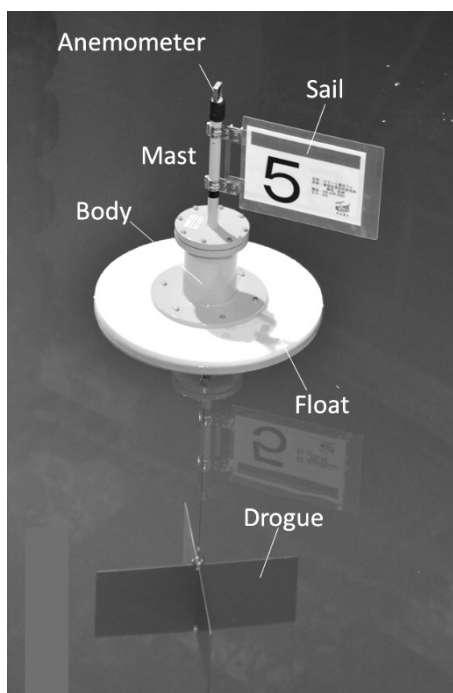


写真-1 ブイの外観

図中のキャプションは図-2を参照

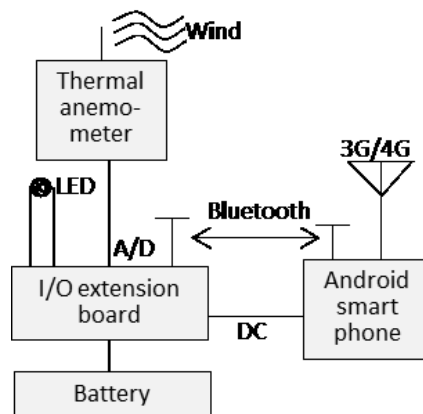


図-3 ブイの電子回路構成

Thermal anemometer: 熱式風速計

I/O extension Board: 拡張基板

Android smart phone: アンドロイドスマートフォン

Battery: 乾電池

## 2.3 電子回路構成

ブイの胴体内部に組み込んだ制御計測システムの概要を図-3に示す。スマートフォンにはアンドロイド携帯電話(Nexus5 Android 4.4 シムフリー版)を使用した。風向風速を計測する機能はスマートフォンには搭載されていないので、この部分については、新たな電子回路を追加した。風速計からは風速に従って変化する電圧が出力されるので、それをA/D変換してスマートフォンに読み込むためにI/O拡張ボード(SPARKFUN IOIO DEV-10748 表-1参照)を登載するベース基板を作成した(図-4)。IOIOはA/D以外にもD/A, PIO, UART, PWMなどの各種I/O機能を提供しており、例えばUARTを利用すればイリジウムによる衛星通信機能などを付加することも比較的容易にできるが、ここでは風速計からの電圧入力と、バッテリー電圧のモニタリング用の電圧入力の二つの信号のみを取り込むことにした。電圧入力においては、念のためIOIOの入力端子前にオペアンプによるボルテージフォロワ回路並びに過電圧保護回路を設けた。さらには出力としてブイのマストに搭載するLED点灯用にオープンドレイン回路を設けた。拡張ボードとスマートフォンとの接続にはBluetoothインターフェースを使用した。2012年の一次試作においては、旧版のIOIOを用いて、スマートフォンとはUSBにより直接接続していたが、IOIOからスマートフォンへの給電能力に限界があったことと、ソフトウェア開発時にスマートフォンのUSB端子は実機デバッグ用にパソコンに繋げる必要があることから、2013年版では、IOIOとスマートフォン間はBluetoothによる接続を採用することとした。電源には乾電池を用いたが、円筒容器への収まりの良さから単三型を採用した。8本を直列接続

し、それを最大6系統並列で実装できるようにした。電池から供給される電源はDC-DCコンバータによりDC5Vに変換し、風速計、並びにスマートフォンの外部電源として与えた。ブイ内部の実装は写真-2に示すように、下部に質量の大きい電池を束ねて収め、上部に電子回路及びスマートフォンを収納している。

乾電池として使い切りタイプの高性能アルカリ乾電池(LR6EJN)とニッケル水素の充電式乾電池(BK-3MCC)を試験した。最大搭載量の半分、即ち3系統並列で稼働させた場合の電池の消耗実験結果を図-5に示す。図の2本の曲線は乾電池の電圧値の変化であり、上の2本の折れ線はスマートフォンの内部電池の残量表示である。スマートフォンが内部電池を消費し始めるまで時間を一応の目安にすると、ニッケル水素充電式乾電池で約18時間(36時間 乾電池最大搭載時換算)、使い切りタイプの高性能アルカリ乾電池で約22時間(約44時間 乾電池最大搭載時換算)の連続稼働が可能であることが分かった。都度回収する日中の洋上試験では十分な時間であるし、最大量の電池を搭載すれば昼夜を超えた実験や監視も可能であることがわかる。



Circuit side Smartphone side

写真-2 ブイ内部の電子機器

Circuit side:回路側, Smartphone side:スマートフォン側

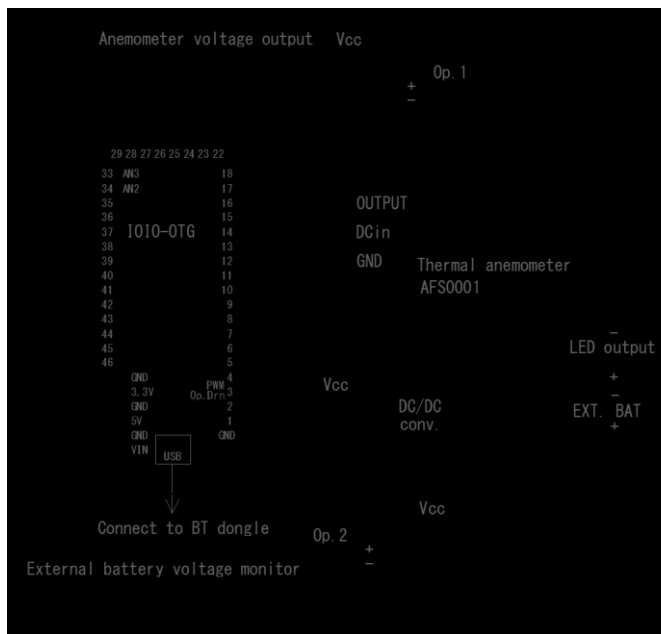


図-4 外部入出力用拡張ボード回路

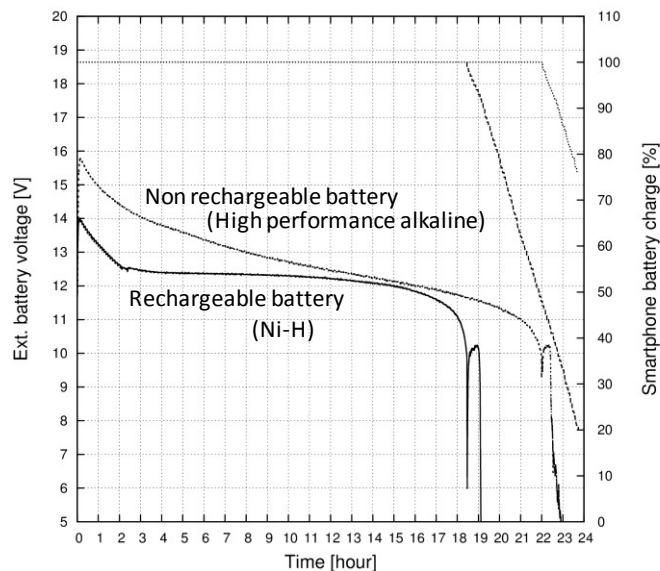


図-5 バッテリー消耗試験

縦軸(左):外部電池電圧[V],縦軸(右):スマートフォン内部バッテリー充電量[%],Rechargeable battery:充電式乾電池,Non-rechargeable battery:非充電式乾電池



## 2.4 ソフトウェア構成

### (1) ブイの計測通信制御プログラム

ブイに搭載するスマートフォンのソフトウェアの開発は Linux PC 上で統合開発環境の Eclipse を用い Java で行った。ソフトウェアの構成を図-6 に示す。アンドロイドスマートフォンのアプリケーションは通常画面タッチをマンマシンインターフェースにし、アプリケーションに対し動作の指示を与える。通常こうしたアプリケーションは Java のクラスにおいて Activity クラスを継承したクラスとして構築される。しかし組み込み機器としての応用を考えた場合には、Activity クラスは長時間の画面タッチ入力がない場合の永続的な動作が必ずしも保証されないという問題点があるので、今回の開発では、アプリケーションの起動部分には Activity クラスを継承したクラスを用い、その後の永続的な動作が期待される部分には画面を持たず、いわばバックグラウンドジョブとして永続的に動作を続ける IntentService クラスを継承するクラスとして、各種の機能を実現した。主なクラスとしては、LAN 回線を通じてユーザからのコマンド等を受け取り、対話を行うための TCPService、IOIO 拡張ボードを制御するための IOIOService4buoy クラス、GPS 情報などスマートフォンに内蔵されているセンサの情報を扱う GPS2Logservice クラス、計測情報を無線回線で送信する LOGservice クラスなどを用意した。IOIOService4buoy クラスや GPS2Logservice クラスなどで得られたデータは一旦、アンドロイド OS のシステムログに書き出し、LOGservice において再度読み込んで、UDP プロトコルを用いてインターネットへ送信する。UDP は TCP と異なりコネクションレスプロトコルで、送信されたパケットが相手先に届いたかどうかの確認を行わない。そのため通信に不確実性があるが、逆にシンプルなプロトコルであり、高速な通信が可能である。本開発では速度重視の観点から UDP を採用している。

LOGservice クラスの一部抜粋をリスト-1 に示す。アンドロイドのシステムコマンドである logcat を exec () コマンドによりアプリケーションから呼び出すことで、システムログからデータと読み込み SD カードあるいは UDP パケット通信を実現している。システムログへの書き出し側の例として、GPS2Logservice クラスの一部抜粋をリスト-2 に示す。このクラスでは、GPS センサに関する機能と、内部センサに関する動作を定義しているが、その為にクラスの定義で LocationListener と SensorEventListener のインターフェースを実装している。例えば GPS のロケーション情報に変化が生じた場合、onLocationChanged()メソッドが呼び出されるので、メソッドにおいて新しいデータを読み込み、Log.i() コマンドによりシステムログに書き出す。

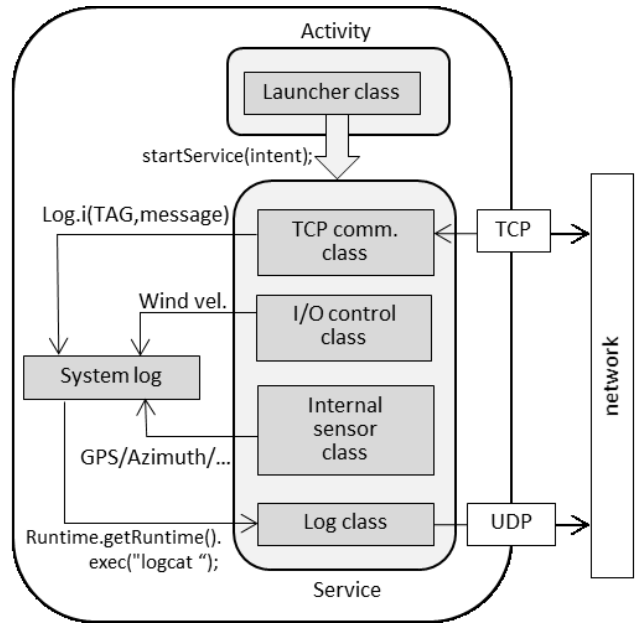


図-6 ソフトウェア構成

リスト-1 LOGservice 抜粋

```

////////////////////////////////////
public class LOGservice extends IntentService
////////////////////////////////////
{
    protected void onHandleIntent(Intent intent) {
        //logcatを呼び出す。
        proc=Runtime.getRuntime().exec(
            "logcat -v time "+LOGTAG+":!");
        //logcatに対するBufferedReaderを定義する。
        logreader=new BufferedReader(
            new InputStreamReader(
                proc.getInputStream()),1024);
        while(true){
            while((line=logreader.readLine())!=null){
                //SDカードへの書き出し
                sd.write(line+"¥n");
                //UDP プロトコルでの送信
                //ソケットを開く
                DatagramSocket udpSocket = new
                    DatagramSocket();
                //パケット作成
                DatagramPacket packet = new DatagramPacket(
                    (line+"¥r¥n").getBytes(),
                    line.length()+2,
                    InetAddress.getByName(IP),
                    UDP_PORT);
                //送信先アドレスとポート
                //パケット送信
                udpSocket.send(packet);
                //ソケットを閉じる
                udpSocket.close();
            }
        }
    }
}

```

リスト-2 GPS2Logservice 抜粋

```

////////////////////////////////////
public class GPS2Logservice extends IntentService
implements LocationListener, SensorEventListener
////////////////////////////////////
{
    private double mag_x,mag_y,mag_z,azimuth;

    protected void onHandleIntent(Intent intent){
        while(!flag_STOP){
            //1000msec おきに方位角を書き出す
            Thread.sleep(1000);
            Log.i(LOGTAG, "<AZIMUTH/MAG>"+(float)azimuth+ " "
                +(float)mag_x  + " "
                +(float)mag_y  + " "
                +(float)mag_z);
        }
        .....;
    }

    //////////////////////////////////////
    //インターフェースの実装
    //LocationListener SensorEventListener
    //////////////////////////////////////

    //ロケーション情報が変更された時に呼び出される
    public void onLocationChanged(Location location) {
        //GPS 情報
        String lat = String.valueOf( location.getLatitude());
        String lon = String.valueOf( location.getLongitude());
        String acc = String.valueOf( location.getAccuracy() );
        String alt = String.valueOf( location.getAltitude() );
        String speed=String.valueOf( location.getSpeed());
        String time= String.valueOf( location.getTime());
        String bear= String.valueOf( location.getBearing());
        //ログ情報作成
        logmessage ="<GPS>" + lat+" "
            + lon+" "
            + alt+" "
            + acc+" "
            + bear+" "
            + speed+" "
            + time;

        //システムログへの書き出し
        Log.i(LOGTAG,logmessage);
    }

    //センサの状態が変化した時に呼び出される
    public void onSensorChanged(SensorEvent event)
    {
        if(event.sensor.getType()==
            Sensor.TYPE_MAGNETIC_FIELD){
            mag=event.values.clone();
            mag_x=mag[0];
            mag_y=mag[1];
            mag_z=mag[2];
            mag_r=Math.sqrt(mag_x*mag_x+mag_z*mag_z);
            azimuth=Math.atan2(mag_x,mag_z)
                *180.0/3.141592+180.0;
        }
    }
    .....;
}

```

リスト-3 IOIOservice4buoy 抜粋

```

////////////////////////////////////
public class IOIOservice4buoy extends IOIOService
////////////////////////////////////
{
    final static String TAG_HEAD="<AUX";
    private String LOGTAG;
    @Override
    protected IOIOLooper createIOIOLooper() {
        return new BaseIOIOLooper() {
            private DigitalOutput led_;
            private AnalogInput an0_,an1_,an2_;
            private PwmOutput pwm3;
            float an0value,an1value;
            float windvelocity, extbatvolt;

            @Override
            protected void setup() throws
            ConnectionLostException, InterruptedException
            {
                //電圧入力端子を定義
                an0_ =ioio_.openAnalogInput(Const.AN0);
                an1_ =ioio_.openAnalogInput(Const.AN1);
                //出力端子を定義して PWM 出力
                pwm3=ioio_.openPwmOutput(new
                    DigitalOutput.Spec(3,Mode.OPEN_DRAIN),1);
                pwm3.setDutyCycle((float) 0.5);
            }

            @Override
            public void loop() throws ConnectionLostException,
            InterruptedException
            {
                Thread.sleep(1000); //1000msec 休む
                //風速計
                an0value = an0_.getVoltage();
                windvelocity=(float) (4.94566*
                    Math.pow(an0value-1.80,3.26542));
                Log.i(LOGTAG,TAG_HEAD+":WIND>"
                    +an0value+ " " +windvelocity); //ログに出力
                //外部バッテリー電圧
                an1value = an1_.getVoltage();
                extbatvolt=(float) (an1value*5.0); // 1:5 に分圧
                Log.i(LOGTAG,TAG_HEAD+
                    ":EXT.BAT>"+extbatvolt);
            }
        };
    }

    @Override
    public void onStart(Intent intent, int startId) {.....}
}

```

外部拡張ボードの制御クラスである IOIOService4buoy クラスの概要をリスト-3 に示す。IOIO ボードのメーカーが提供するクラス IOIOService クラスを継承する形で定義している。IOIOService クラスを利用するためにはライブラリを開発環境にインストールする必要があるが、方法等は GitHub の IOIOLib Basics に詳述されている<sup>7)</sup>。本資料の場合、リストの setup()メソッドにおいて入力として電圧を 2 チャンネル、出力としてデジタル出力を 1 チャンネル定義している。出力チャンネルに対しては 1Hz でデューティ比



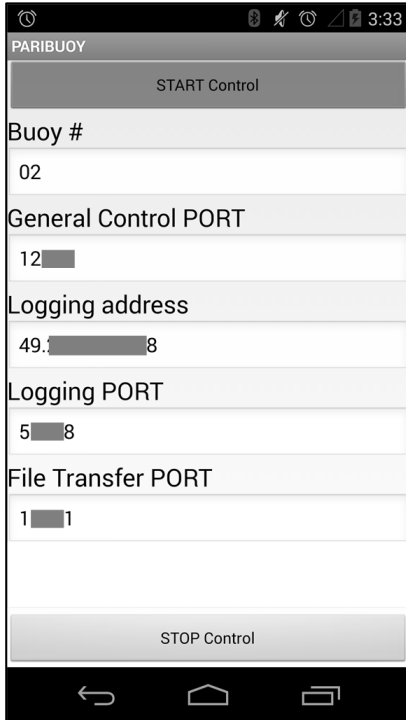


写真-3 アプリケーション始動画面

リスト-5 Launcher アクティビティ抜粋

```

////////////////////////////////////
public class Launcher extends Activity {
////////////////////////////////////
//START ボタンが押された場合の処理
startButton.setOnClickListener(
new OnClickListener(){
@Override
public void onClick(View v) {
//ログサービス用に新しいインテントを作成
Intent intent=new Intent();
//IP アドレス情報をセット
intent.putExtra("LOGGING_ADDRESS",
LoggingAddress);
//ポート情報をセット
intent.putExtra("LOGGING_PORT",
LoggingPort);
//サービス名情報をセット
intent.setClassName(Const.PACKAGE,
Const.CLASS_LOG);
intent.setAction("FILTER_ON");
//サービスを起動
startService(intent);
//ログに起動したことを出力
Log.i(LOGTAG,"<LAUNCHER>LOG START");
..... ;
//他のサービスを起動
..... ;
finish();
}
});
.....;
}

```

リスト-6 UDP2stdout.c

```

////////////////////////////////////
// UDP2stdout.c
// which receives data from UDP PORT
// and print them to stdout.
// usage: UDP2stdout PORT#
////////////////////////////////////
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

int main(int argc,char* argv[])
{
int sock;
struct sockaddr_in addr;
struct sockaddr_in senderinfo;
socklen_t addrlen;
char buf[2048];
int n;
int PORT;
PORT=atoi(argv[1]);
//データグラムソケットを作る
sock = socket(AF_INET, SOCK_DGRAM, 0);
addr.sin_family = AF_INET;
addr.sin_port = htons(PORT);
addr.sin_addr.s_addr = INADDR_ANY;
bind(sock,
(struct sockaddr*)&addr,
sizeof(addr));
addrlen=sizeof(senderinfo);
//データの読み込みと書き出し
while(1){
n=recvfrom(sock,
buf,
sizeof(buf)-1,
0,
(struct sockaddr*)&senderinfo,
&addrlen);
write(fileno(stdout),buf,n);
}
close(sock);
return 0;
}

```

(2) サーバ側プログラム

漂流ブイは UDP プロトコルによるログデータの配信を行うのでデータを受け取る側はこれに対応したサーバ機能が必要になる。UDP に関する受信プログラムの例はインターネットでも多く公開されているが、ここでは C 言語で書いた簡単な例(UDP2stdout.c)をリスト-6 に示す。このプログラムは例えば Linux 環境であれば、コマンドプロンプトから

```
[hoge@hoge ~]$ UDP2stdout 12345 | tee Logfile
```

と実行することで、ポート番号12345をUDPの受信信用に開いて、そこに送られて来るデータを標準出力とLogfileに書き出すことができる。漂流ブイから送られてくるログデータの例を図-9に示す。これは海上での試験における実データの一部で、データの並びはタイムスタンプに続いてブイの番号<計測項目>計測データという構成になっている。漂

流ブイとサーバ間の通信にUDPを用いる場合、サーバ側は通信相手とのコネクションを特に関知しないため、複数の漂流ブイのデータをひとつのポートで受け取ることができ

る。この例では5台の漂流ブイからのデータを受け取っている。

```

12-28 14:54:08.446 I/PARI_BUOY_05( 3219): <GPS>35.31436446,139.36597029,46.0,3.0,0.0,0.0,1388210053000
12-28 14:54:14.284 I/PARI_BUOY_03( 3231): <GPS>35.31433273,139.36597809,35.0,3.0,0.0,0.0,1388210053000
12-28 14:54:12.412 I/PARI_BUOY_01( 2902): <GPS>35.3143431,139.36598098,37.0,5.0,0.0,0.0,1388210053000
12-28 14:54:07.747 I/PARI_BUOY_04( 3179): <GPS>35.31435485,139.36596698,35.0,4.0,0.0,0.0,1388210053000
12-28 14:54:09.194 I/PARI_BUOY_02( 3206): <GPS>35.31438114,139.36598895,40.0,10.0,0.0,0.0,1388210053000
12-28 14:54:09.284 I/PARI_BUOY_02( 3206): <AUX:WIND>2.1 0.09700745
12-28 14:54:09.284 I/PARI_BUOY_02( 3206): <AUX:EXT.BAT>12.483871
12-28 14:54:12.582 I/PARI_BUOY_01( 2902): <AZIMUTH/MAG>76.9054 -28.797913 -28.677368 -6.6986084
12-28 14:54:12.582 I/PARI_BUOY_01( 2902): <ACCELERATION>0.5446167 9.820496 -0.009292603
12-28 14:54:09.464 I/PARI_BUOY_02( 3206): <AZIMUTH/MAG>94.42815 -31.999207 -33.251953 2.4780273
12-28 14:54:09.464 I/PARI_BUOY_02( 3206): <ACCELERATION>-0.14154053 9.693634 0.17727661
12-28 14:54:12.812 I/PARI_BUOY_01( 2902): <AUX:WIND>2.2290323 0.31199646
12-28 14:54:08.147 I/PARI_BUOY_04( 3179): <AZIMUTH/MAG>101.50162 -26.38092 -33.052063 5.368042
12-28 14:54:12.812 I/PARI_BUOY_01( 2902): <AUX:EXT.BAT>12.370968
12-28 14:54:08.147 I/PARI_BUOY_04( 3179): <ACCELERATION>0.049987793 9.80452 -0.11552429
12-28 14:54:14.744 I/PARI_BUOY_03( 3231): <AUX:WIND>2.1032257 0.100455254
12-28 14:54:14.744 I/PARI_BUOY_03( 3231): <AUX:EXT.BAT>12.435484
12-28 14:54:14.874 I/PARI_BUOY_03( 3231): <AZIMUTH/MAG>110.31646 -32.484436 -33.2489 12.026978
12-28 14:54:08.367 I/PARI_BUOY_04( 3179): <AUX:WIND>2.2096775 0.26833963
12-28 14:54:14.874 I/PARI_BUOY_03( 3231): <ACCELERATION>-0.0058898926 9.959595 -0.1775055
12-28 14:54:08.367 I/PARI_BUOY_04( 3179): <AUX:EXT.BAT>12.564516
12-28 14:54:09.226 I/PARI_BUOY_05( 3219): <AUX:WIND>2.0290322 0.04018073 //風速情報
12-28 14:54:09.226 I/PARI_BUOY_05( 3219): <AUX:EXT.BAT>12.612904 //外部バッテリー電圧
12-28 14:54:09.236 I/PARI_BUOY_05( 3219): <AZIMUTH/MAG>92.996544 -24.601746 -37.265015 1.2878418 //方位角情報
12-28 14:54:09.246 I/PARI_BUOY_05( 3219): <ACCELERATION>0.08041382 9.823578 0.36094666
12-28 14:54:08.727 I/PARI_BUOY_04( 3179): <GPS>35.31435486,139.36596698,35.0,4.0,0.0,0.0,1388210054000
12-28 14:54:10.204 I/PARI_BUOY_02( 3206): <PHONE SIGNAL STRENGTH> -83dBm 15 asu //電波強度
12-28 14:54:15.284 I/PARI_BUOY_03( 3231): <GPS>35.31433273,139.36597809,35.0,3.0,0.0,0.0,1388210054000 //GPS 位置情報
12-28 14:54:13.422 I/PARI_BUOY_01( 2902): <GPS>35.31434241,139.36598143,37.0,5.0,0.0,0.0,1388210054000
12-28 14:54:09.486 I/PARI_BUOY_05( 3219): <GPS>35.31436445,139.36597028,46.0,3.0,0.0,0.0,1388210054000
12-28 14:54:10.214 I/PARI_BUOY_02( 3206): <GPS>35.31438113,139.36598895,40.0,10.0,0.0,0.0,1388210054000

```

タイムスタンプ      ブイ番号      <計測項目> 計測データ

図-7 ログデータの例

取得したデータファイルからリスト-7に示す簡単なシェルスクリプトを使ってkmlファイルを作ることができる。リストの下線部分で、grep、sedやgawkなどのLinuxのコマンドをパイプでつなげてログファイルからkmlファイルに必要な位置情報を抽出している。得られたkmlファイルをGoogle earthで表示した例を図-8に示す。



図-8 kml ファイル表示 (Google earth)

### リスト-7 kml への変換スクリプト

```
#!/bin/bash
#####
# KML converter Log2kml.bash
# Usage: Log2kml.bash logfile buoynum
#####
LOGFILE=$1
BUOYNUM=$2

cat <<EOF
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
<name>PARI_BUOY</name>
<description> DRIFTING BUOY TEST </description>

<Style id="Brown">
  <LineStyle>
    <color>50143c96</color>
    <width>5</width>
  </LineStyle>
  <PolyStyle>
    <color>50143c96</color>
  </PolyStyle>
</Style>

<Style id="Yellow">
  <LineStyle>
    <color>5014f0ff</color>
    <width>5</width>
  </LineStyle>
  <PolyStyle>
    <color>5014f0ff</color>
  </PolyStyle>
</Style>

<Placemark>
<name>BUOY ${BUOYNUM}</name>
<description>
  Transparent green wall with yellow outlines
</description>
<styleUrl>#Yellow</styleUrl>
<LineStyle>
<extrude>1</extrude>
<tessellate>0</tessellate>
<altitudeMode>absolute</altitudeMode>
<coordinates>

$[grep "BUOY ${BUOYNUM}.*<GPS>" $LOGFILE |
sed -e "s/\^.*>||g"|
gawk 'BEGIN{FS=" "}{print $2","$1"."$3}'|
tail -n 36000 ]

</coordinates>
</LineStyle>
</Placemark>

</Document>
</kml>
EOF
```

### 3. 簡易漂流テレメトリブイの実海域試験と流出油捕捉

#### 3.1 機能試験

試作したスマートフォン内蔵漂流ブイは実海域において

機能並びに運用性に関して試験を実施した。2012年度は横須賀市金田湾沖, 2013年度は平塚沖海域において実施した。実施海域の場所を図-9に示す。図にはNTTdocomoの利用可能エリアも示してある。海上であっても比較的広い範囲をカバーしており, 沿岸域での用途であれば, 携帯電話の無線回線が利用できる可能性を示している。

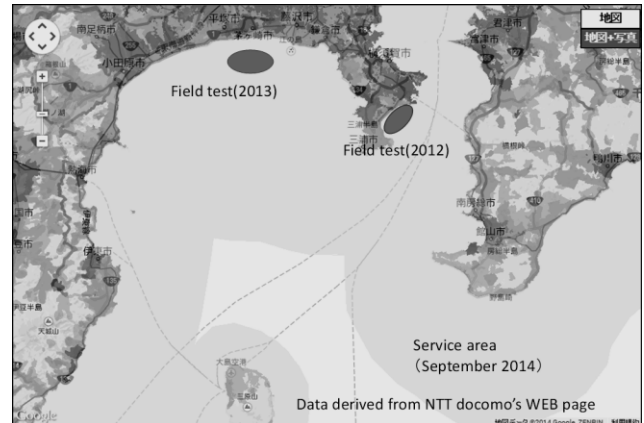


図-9 実海域試験海域  
Field test:実海域試験, Service area:通信範囲

携帯電話のパケット通信に必要となるSIMについてはNTTdocomoのMVMOのSIMを使用した。2013年度に実施したネットワークの構成を図-10に示す。漂流ブイとユーザ端末(PCあるいはタブレット)間のデータの中継を行う為に、インターネット上にVPS (Virtual Private Server, メモリ1GB, ディスク容量100GB, CPU: 仮想2コア, OS: Linux CentOS 6) を用意した。VPS上にUDPレシーバを動作させ、漂流ブイからのデータを受け取り、ログファイルに保存するようにした。さらに受信データをVPSに対してPPTP (Point to Point Tunneling Protocol) によりVPN (Virtual Private Network) 接続されたユーザの端末に逐次転送し、写真-4に示す様に、ユーザ端末から漂流ブイの状況(測位位置, バッテリ電圧, 電波状況, 風速値など)を常時リアルタイムで監視できるようにした。2012年度の試験では漂流ブイとVPSの間にもPPTPによるVPN接続を行ったが、VPN接続が切れた場合の復帰処理に難しさがあり、2013年度の試験においては漂流ブイとVPS間にはVPN接続は使用していない。

金田湾沖で実施した2012年度の試験では、通信関連の基本的な動作、ブイの形状の検討などを行った。当初の試作一号機ではブイの周囲の浮き輪状の浮体は取り付けなかった。この場合写真-5(a)に示す様に波浪中ではブイ全体が大きく動揺し、風向風速がうまく計測できなかったことから、前述の様にブイの胴体へのリング状の浮き輪を追加

し、動揺を抑え、マストの垂直安定性の改善を図った。2013年12月26日に行った試験におけるブイの様子を写真-5(b)に示す。この日は13時過ぎから風向きが南からに変わり、風速は6m/s前後になっていた。これに伴い実験海域では沖から波浪が襲来し、海面状況はかなり乱れていた。しかしながら写真からもわかるように、ブイは追加したリング状の浮体により安定した姿勢を保っていた。

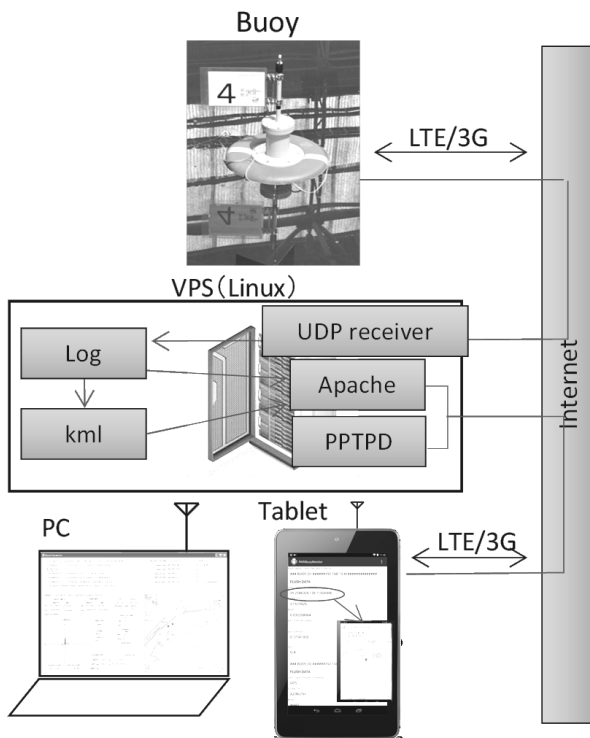
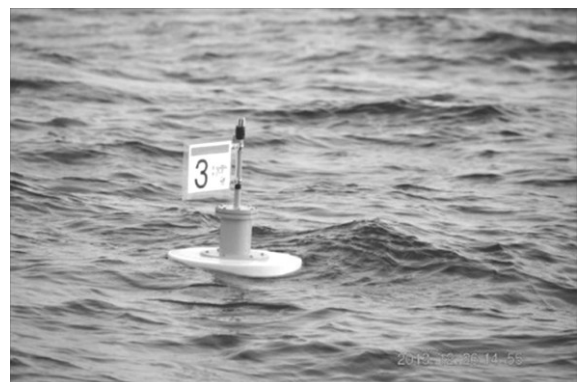


図-10 実海域試験におけるネットワーク構成



(a)一次試作ブイ



(b)二次試作ブイ

写真-5 洋上におけるブイの姿勢

ブイに組み込んだスマートフォンの実海域における電波の受信強度の変化を図-11に示す。横軸が時刻であり、縦軸は電波強度を示している。図中でレベル分けしているのは、携帯電話の上部に図案で表示されるアンテナの数の参考であり、ここではNTTdocomoの場合を例示している。図の例は2013年12月27日におけるもので、5台のブイを投入している。9時から14時の間が洋上での漂流状態であり、それ以外は船の上での値である。洋上での漂流は沖合1.3~1.9kmの範囲にある。14時以降ブイは伴走船に回収されているが、船上での最大離岸距離は3.5km前後であった。電波強度はブイが船上置かれた場合と比較して、洋上に投入された場合、十数dBm下がることになる。他の日でも同様の傾向が見られた。電波の受信感度はスマートフォンが設置される水面からの高さに影響を受けている。大島ら<sup>8)</sup>は東京湾内において3G回線の電波強度調査を実施し、浮体に通信機器を仕込んだ場合と海面から10m上に設置した場合で比較しているが、著者らと同様の傾向が見られている。このように電波強度の低下は見られたものの、今回の平塚沖における複数日、複数台による実験においては、-130dBm以下に電波強度が下がり通信不能になることはなかった。各々のブイ

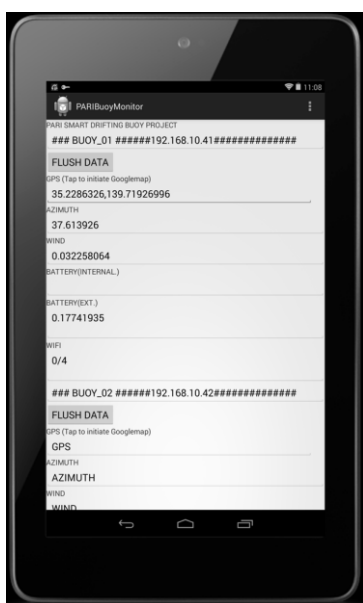


写真-4 ブイの監視アプリ

は毎秒400Byte程度のデータを常時通信で送ってくるように設定していたが、それらが途切れることはなかった。

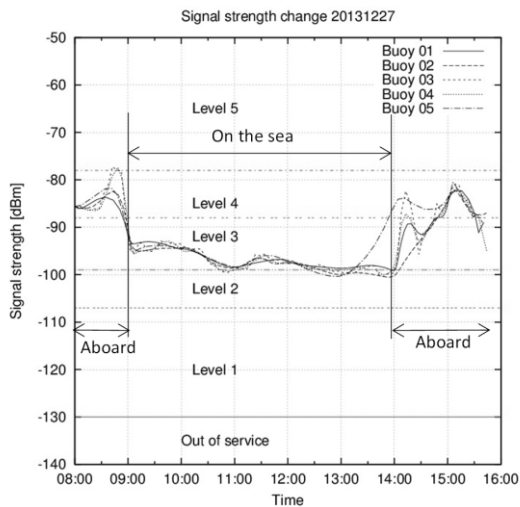


図-11 実海域試験における電波強度  
縦軸:信号強度[dBm],横軸:時刻[時],  
Aboard:船上,On the sea:海上,  
Level #:電波レベル,凡例:ブイ番号

次に風向風速測定に関して述べる。漂流ブイでの風向風速計測はマストの頂上に取り付けた熱式の風速計とスマートフォン内蔵の磁気センサを併用して行われる。風速計は写真-6に示す形状をしており、発熱した白金抵抗体部分の、温度による抵抗値の変化から風速を検出している。一方風向きはブイのマストに取り付けた尾翼が風に対する抵抗になり、ブイ全体が風上を向くように工夫したことで、スマートフォンの磁場センサを使って風向を検出できるようになっている。スマートフォンの磁場センサの座標系は図-12に示すようになっており、スマートフォンを縦に(y軸を上方に向けて)設置した場合、図-13に示すように、x方向とz方向の磁場の位相関係より方位を検出することができる。こうした風向風速計測の仕組みが実海域においても正しく機能するかを確認した。図-14は2012年度に実施した実海域試験における伴走船に設置した超音波風向風速計による風速値とブイによる風速値の比較であり、ブイには前述したリング状の浮体を追加している。風速値は両者で比較的良く一致している。また図-15は風向を超音波風向風速計とブイで比較したもののだが、これも良く一致している。図-16は2013年度の試験結果で5台投入したブイの風速を比較したもので、大きな個体差は認められない。また洋上で複数のブイを観察していると、写真-7に示すように、尾翼の向きは概ねどれも揃っており、洋上においても正しい風向を向いていることが目視でも観察できた。

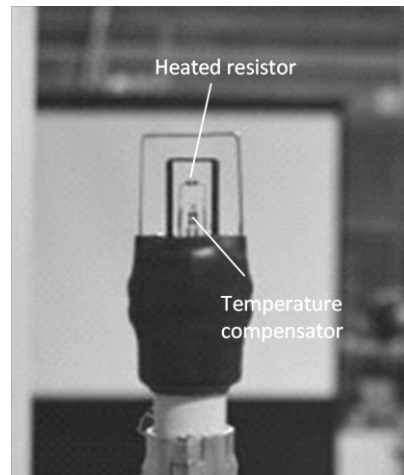


写真-6 風速計  
Heated resistor:発熱抵抗体,  
Temperature compensator:温度補償素子

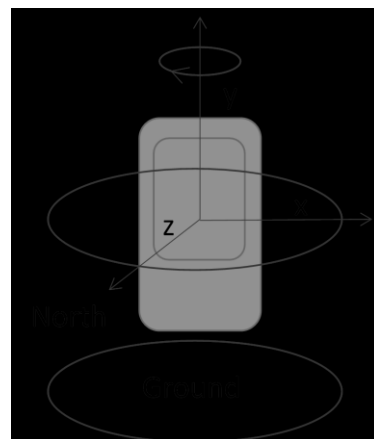


図-12 スマートフォンの内部センサの座標系  
Ground:地表, North:北

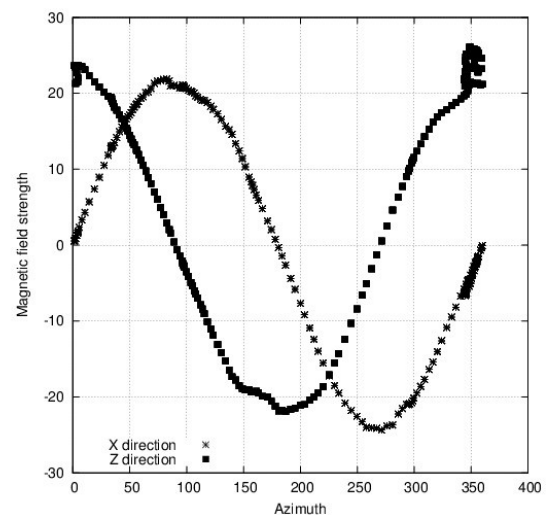


図-13 内部磁場センサによる方位角計測  
縦軸:磁気強度,横軸:方位角[°],X direction:X軸方向,  
Y direction:Y軸方向



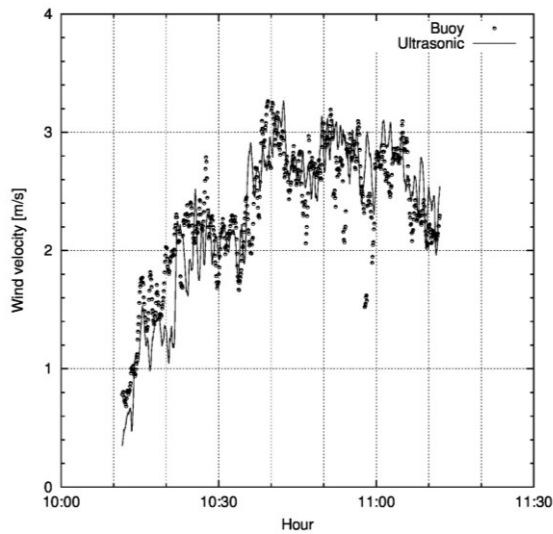


図-14 ブイと超音波風向風速計における風速の比較  
縦軸:風速[m/s],横軸:時刻[時],Ultrasonic:超音波風向風速計による測定値,Buoy:ブイによる測定値

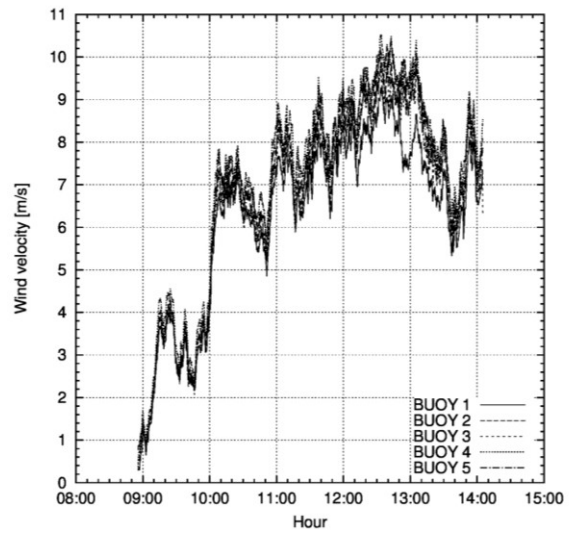


図-16ブイにおける風速の比較 (2013年度)  
縦軸:風速[m/s],横軸:時刻[時],凡例:ブイ番号

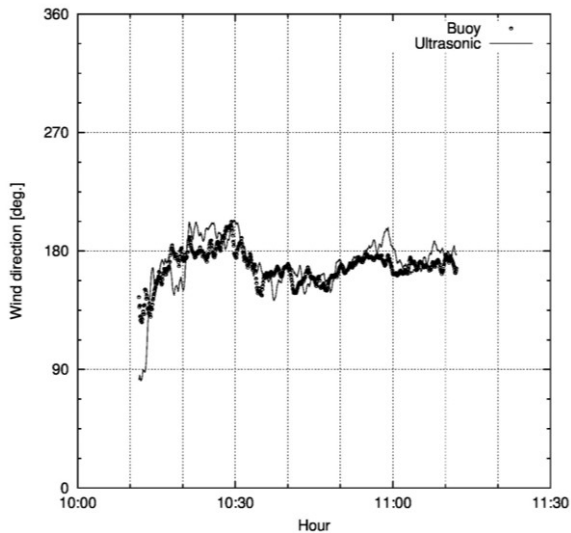


図-15 ブイと超音波風向風速計における風向の比較  
縦軸:風向[°],横軸:時刻[時],Ultrasonic:超音波風向風速計による測定値,Buoy:ブイによる測定値



写真-7 実海域でのブイの様子

次に実海域におけるバッテリーの消費量について述べる。実海域試験におけるブイのバッテリー消費量に関するデータを図-17に示す。図は2013年12月27日のブイの結果である。この時ブイは高性能アルカリ乾電池(LR6EJN)を24本搭載していた。およそ8時間の連続運転において、電圧値は12.4Vまで低下した。事前に実施した実験室でのテストでは8時間運転した後にも13V前後の電圧を維持していたことを考えると、海上で且つアンテナ高度が低く、電波状況の悪い場所では電力の消費が若干激しい可能性も考えられるが、最大では倍の48本のバッテリーを搭載できることを考えれば、日中に行う漂流実験、あるいは夜間の漂流油の捕捉などの用途に関しては十分な運転持続時間を有していることが確認できた。

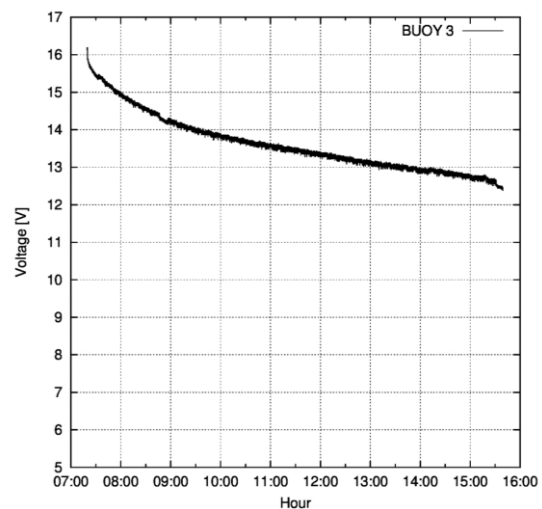


図-17実海域試験におけるバッテリーの消耗  
縦軸:電圧[V],横軸:時刻[時]

### 3.2 漂流油拡散試験

海上流出油の漂流モデルを考える上では、移流と拡散の両方を正確にモデル化する必要がある。拡散に関しては、Spreadingと呼ばれる流出油の物性（海水との密度差、表面張力差、粘性等）に起因した拡散と流れ場の乱れに起因した乱流拡散の二つを考慮しなければならない。Spreadingに関しては、古典的なモデリングとしてFay<sup>9)</sup>のものが知られており、その後いろいろな研究者により修正が図られている<sup>10)</sup>。著者らの開発した流出油の漂流予測シミュレータにおいては、油を粒子の集合と見立てた粒子拡散モデルを用いているが、そこでもFayのモデルを合うように粒子間に距離に従って反発力を与えるモデルを使用している。一方、乱流拡散については、拡がりのスケールと拡散係数との関係でリチャードソンの4/3乗則などが知られているが、定性的なものであり、流出油の水平乱流拡散のモデルとして完成されたものは未だ見られないことから、シミュレーションなどでは、経験に基づいて1-100m<sup>2</sup>/sの範囲で水平乱流拡散係数を与えることが多い<sup>11)</sup>。更に、流出油は大気と海水の間の気液界面に非常に薄い層を形成するが、一般的に表層流と称される、それでもある程度の厚さを持った海面近傍の水塊とは異なる運動をすることが考えられる。著者らはこれまで風による油の運動の過渡応答特性<sup>12)</sup>や、ゴムマットを用いた実海域での漂流拡散実験<sup>13)</sup>を実施し、浮遊油の水平乱流拡散係数の定量的モデル化に努めている。

しかし実際の海域における気液界面層の水平乱流拡散特性を定量的に把握しモデル化するためには、表層水塊と気液界面層の拡散を同時に計測し、洋上風や海水の速度などとの相関を解析する必要がある。こうした目的に対しては、洋上での風向風速が計測でき、GPS測位が可能な漂流ブイが有効である。

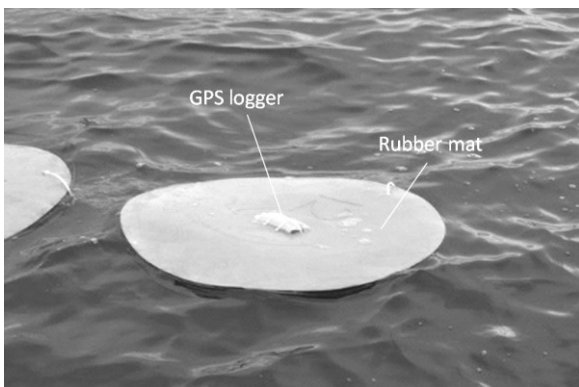


写真-8 擬似油

GPS logger:GPS ロガー, Rubber mat:ゴムマット

実験手法の有効性を検証する為に、2013年度に神奈川県平塚沖において、擬似油としてのゴムマット20枚と漂流ブイ5台を同時投入する漂流拡散試験を実施した。使用したゴムマットは直径1m、厚さ10mm、質量1.2kgで材質はスポンジゴムである。これに漂流位置測定用のGPSロガーを組み込んでいる(写真-8)。漂流ブイには表層水塊の運動に追従するように、抵抗板を水深1.5mの位置に吊り下げた(写真-1)。

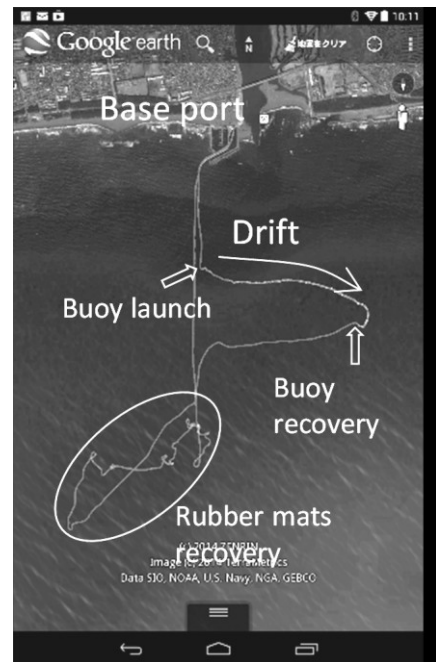


図-18 海上実験における軌跡

Base port:基地港, Buoy launch:ブイ投入,Drift:ブイの漂流軌跡, Buoy recovery:ブイ回収, Rubber mats recovery: ゴムマット回収

一例として12月27日の実験を示す。図-18はkmlファイルによるブイの軌跡のプロットである。ブイの投入時刻は午前9時頃、揚収時刻は午後2時頃である。漂流中のブイの位置と擬似油の位置の比較を図-19に示す。図は30分毎の位置を示している。投入してから回収されるまでの約5時間の間にブイは西から東へ海岸線とほぼ並行に1200m程度移動した。この間、風は北北東の微風から北東風で風速6~9m/sに変化を示した。一方擬似油の漂流軌跡はブイとは大きく異なり、投入後暫くは南方に漂流し、その後北東風に流される形で、漂流方向を南西に変えている。

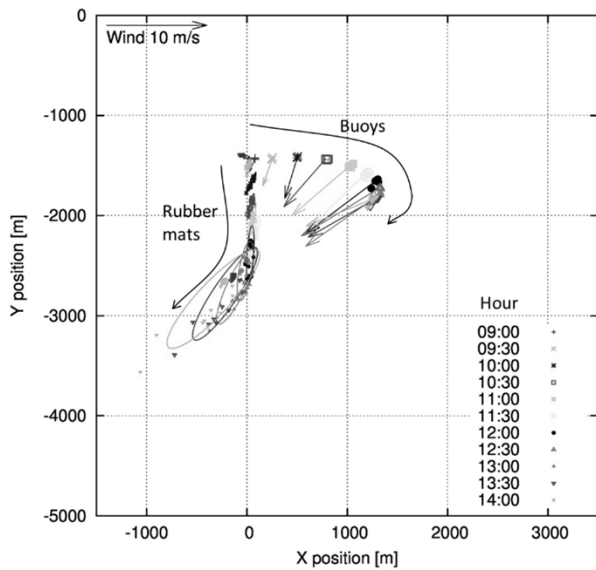


図-19 ブイと擬似油の漂流の比較  
 縦軸:緯度方向距離, 横軸:経度方向距離, Buoy:ブイの漂流位置, Wind vector:風ベクトル, 楕円:ゴムマットの漂流域, 凡例:時刻(秒)

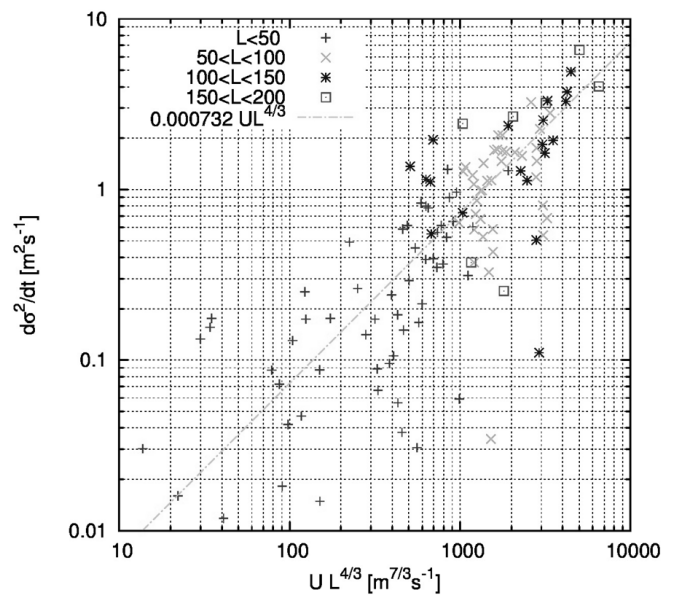


図-21擬似油の拡散係数

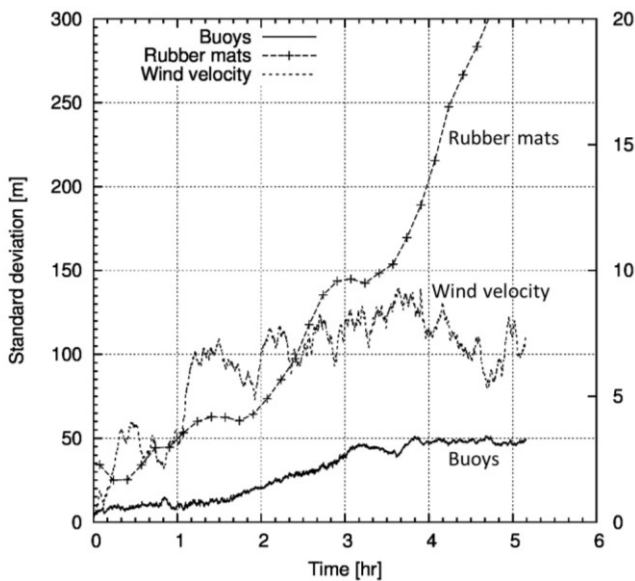


図-20 ブイと擬似油の拡がり方の比較  
 縦軸:標準偏差, 横軸:時刻 Buoy:ブイの拡がり, Rubber mats:ゴムマットの拡がり, Wind velocity:風速

漂流軌跡の違いに加えて、拡がり状況もブイと擬似油で顕著な違いが見られた。擬似油の個数20枚に対して、ブイの投入数が5台と少ないため定量性には疑問の余地があるものの、5台のブイは比較的狭い範囲にかたまわって漂流していたのに対し、擬似油は時間の経過に伴って広い範囲に拡がって分布している様子が観察された。ブイと擬似油各々について、各時刻における重心位置からの距離の分散をプロットしたものを図-20に示す。

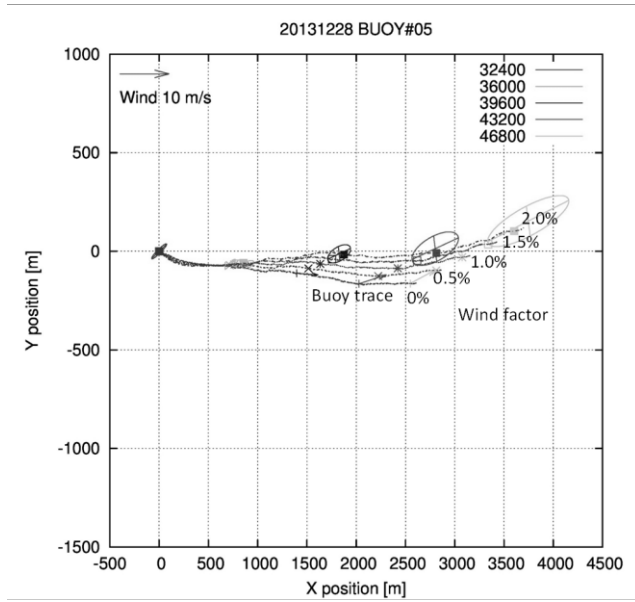
漂流ブイの拡がり方に比べて擬似油の拡がりの方が大きいことがわかる。これは、気液界面層の方が表層近傍水塊に比べて、水平乱流拡散が大きいことを示すものであるが、固体壁面近傍の乱流構造に関する知見などから考えると次の疑問が浮かぶ。気液界面と固体壁では状況は異なるものの、固体壁近傍の乱流構造は一般的に、壁面に接する部分には分子粘性に支配される粘性底層が存在し、境界面から離れるに従って乱流粘性の影響が現れてくることが知られている。この場合、主流から境界面に近づくに従って、乱れは減衰していくが、洋上での大気と海水との境界面において海水側から見るとこのことは成立していない。気液界面層に見られる大きな乱流拡散をもたらす乱れのエネルギーは海水側からではなく、大気側、すなわち洋上風から供給されると考えるのが妥当なように思われる。従って、浮遊油の水平乱流拡散のモデル化にあたっては、洋上風速をパラメータとして組み込む必要があると考えられる。ここでは、試みとして、拡散係数が拡散スケール $L$ の $4/3$ 乗に加えて洋上風に比例すると仮定し、

$$\frac{d\sigma^2}{dt} = cUL^{\frac{4}{3}} \quad (1)$$

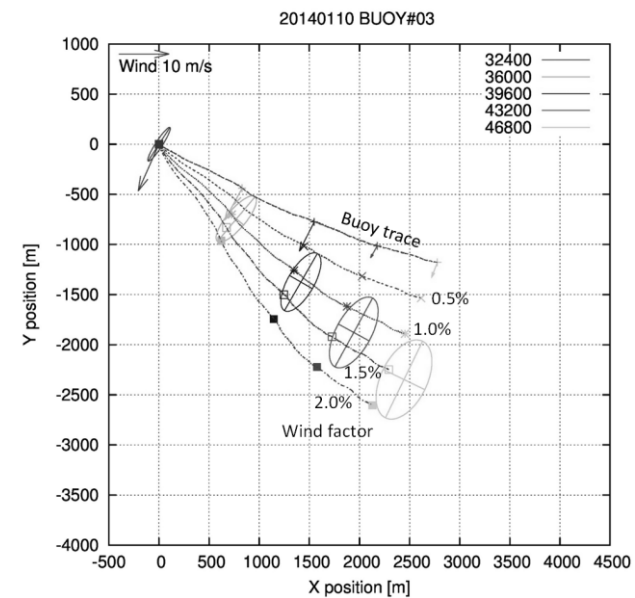
により擬似油の漂流データを整理した。図-21に示す程度の相関が見られた。本資料の主題はブイの開発なので、水平乱流拡散のモデル化については、今後の課題として、高度なモデル化に取り組んでいきたいと考えている。そのためには多くの実海域における実験結果の蓄積が必須であり、本資料で紹介した漂流ブイは有力な計測ツールになると考えている。

### 3.3 漂流油のリアルタイム捕捉の検討

図-19に示した様に漂流ブイと油に見立てたゴムマットの漂流軌跡は一般的には一致しない。これは洋上風によるドリフトの効果がブイとゴムマットで異なっている為である。今回開発したブイではGPSによる位置データに加えて、漂流現場における風向と風速を実時間測定している為、これらの風データを用いてゴムマットの漂流軌跡を推定できるか検討する。



(a) 風が海水流と並行な場合



(b) 風が海水流と直交する場合

図-22 ブイとゴムマットの漂流と風係数  
縦軸:緯度方向距離, 横軸:経度方向距離, Buoy trace:ブイの漂流軌跡, Wind factor:風係数, 楕円:ゴムマットの漂流域, 凡例:時刻[秒]

ブイがある時刻に油と同一地点に投入された場合, 時間 $T$ が経過した後の油の漂流位置は, 風, 流れ共に空間的に一樣であると仮定すれば, ブイの位置と風の時系列を用いて,

$$\mathbf{X}_{obj} = \mathbf{X}_{buoy} + \Delta D \int_0^T \mathbf{U} dt \quad (1)$$

で与えられる。ここで,  $\mathbf{X}$ は位置ベクトル,  $\mathbf{U}$ は風速ベクトル,  $\Delta D$ は風係数の差である。添字の $obj$ は漂流状況を知りたい対象物で, 本来は油で, ここではゴムマット,  $buoy$ は漂流ブイを表す。風係数の差 $\Delta D$ は,

$$\Delta D = D_{obj} - D_{buoy} \quad (2)$$

である。図-22に $\Delta D$ を変えて油の漂流拡散範囲と比較したものを示す。いずれも4時間にわたる漂流実験結果を整理したもので, (a)が海水の流れと風向がほぼ並行な場合で, (b)が海水流と風がほぼ直交する場合である。 $\Delta D$ として1.5~2.0%程度の値を仮定すると, 油の漂流位置を概ね再現できることがわかる。漂流油の風係数は概ね0.03程度であることが知られている<sup>14)</sup>ので, 漂流ブイの風係数 $D_{buoy}$ が既知であれば,  $\Delta D$ が解るので, これにブイの漂流位置に風の影響の時間積分を掛け合わせることで, ブイを基準として油の位置を計算することが可能であることがわかる。

単一のブイを用いた場合にはこの様に, 風の影響の時間積分による補正が必要になる一方, ブイを複数台投入する場合には風の時間積分項を直接計算しないで複数台のブイの漂流位置から推定することが可能になる。今二台のブイが在り, 異なる風係数を有するものとする。二台のブイの相対位置は風係数の違いによるものであり,

$$\Delta \mathbf{X}_{buoy} = \mathbf{X}_{buoy2} - \mathbf{X}_{buoy1} = \Delta D_{buoy} \int_0^T \mathbf{U} dt \quad (3)$$

となる。ここで,  $\Delta D_{buoy}$ は二台のブイの風係数の差であり,

$$\Delta D_{buoy} = D_{buoy2} - D_{buoy1} \quad (4)$$

で定義される。(3)式と(4)式より風の時間積分項は,

$$\int_0^T \mathbf{U} dt = \frac{\Delta \mathbf{X}_{buoy}}{\Delta D_{buoy}} \quad (5)$$

と評価できる。(1)式と(5)式より油の漂流位置は,

$$\mathbf{X}_{obj} = \mathbf{X}_{buoy1} + \frac{\Delta D}{\Delta D_{buoy}} \Delta \mathbf{X}_{buoy} \quad (6)$$

で, 推算することができることになる。(6)式は定性的には漂流物が二台の漂流ブイの位置を結ぶ線分上に存在していることを示している。このことを確認する為に, ブイの下部の抵抗板を外して風係数を大きくしたものを, 抵抗板を装着したブイと同時に流し, ゴムマットの漂流と比較した。

結果を図-23に示す。図は4時間に亘る漂流実験の結果であるが、楕円で描かれている各時刻におけるゴムマットの拡散範囲は、抵抗板を付けたブイと抵抗板を付けないブイの位置を結ぶ直線上に確かに存在しており、(6)式の妥当性を示している。こうしたことが可能である前提条件として、風と流れの空間的な一様性が求められるが、今回の実験の様に4時間程度の比較的短い時間であれば、実験結果から見て一様性の仮定がある程度満たされているように思われる。

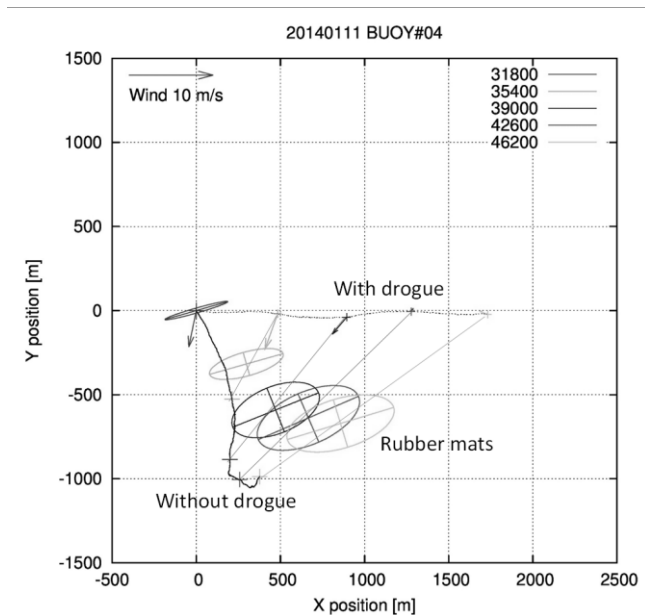


図-23 ブイ二台による漂流位置の推定  
 縦軸:緯度方向距離,横軸:経度方向距離,With drogue:抵抗板を付けたブイの軌跡,Without drogue:抵抗板の無いブイの軌跡, Rubber mats:ゴムマットの拡がり,凡例:時刻[秒]

こうした二台法による推定を行う場合には、ブイに風速計を取り付ける必要はなく、携帯電話単体をブイに組み込めば良く、非常に簡単に装置を構成することが可能である。また組み込むブイも漂流油と同じ風係数を持つように微妙な調整をする必要も無いため、ブイの形状も比較的自由に選ぶことができる可能性がある。二台法においては直接風を計測していないが、風係数の異なるブイの漂流位置の違いから風の影響を推算しているので、油の漂流位置の推定には風の評価が極めて重要であることに変わりはない。

港湾空港技術研究所で開発した流出油の漂流シミュレータは、気象庁の発信する気象、海象情報に関するGPVデータを基にしているが、ここで紹介した洋上のリアルタイム計測データを更に加えることで、油の漂流位置の修正などが行えるようになり、漂流油の予測精度の更なる向上などの効果が期待できる。

## 4. その他の応用

### 4.1 漂流物捕捉ツールへの活用

3章では漂流テレメトリブイによる漂流油の捕捉技術を紹介したが、海上で漂流する大型漂流物にブイを取り付けることで、航路警戒情報を随時取得するといった応用も考えることができる。2014年3月18日未明、神奈川県三浦市の沖合6kmの海上で、パナマ船籍の貨物船BEAGLE3号(12,630t)と韓国船籍PEGASUS PRIME(7,406t)が衝突し、BEAGLE3号が沈没する事故が発生した。この事故により、同船より燃料用のC重油と船倉のハッチが漂流することとなった。漂流したハッチは北西方向に漂流し、図-24に示す千葉県の金谷近郊に漂着している。写真-9に発生した大型漂流物を示す。(a)は海岸線に漂着したハッチであり、(b)は浅瀬で屹立する形で座礁した漂流ハッチである。こうした大型の漂流物が発生すると、船舶への二次衝突の危険を伴うため、回収までの間、常時漂流位置をモニタリングすることが望ましい。図-25に示すように本資料で紹介したブイに索を設け、例えば強力な磁石で対象物に取り付けることで、漂流物にタグ付けし、その後の漂流位置のリアルタイム監視などにも活用できると考える。

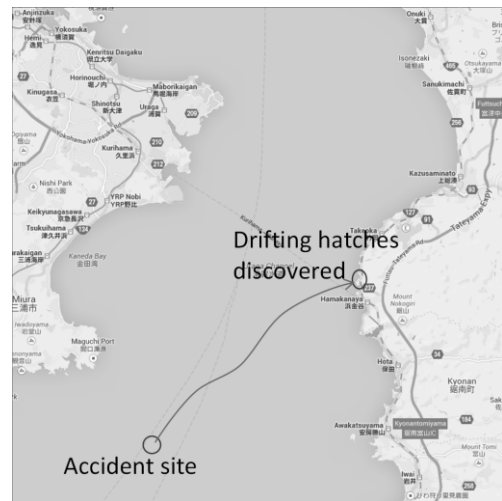


図-24 BEAGLE3号事故における漂着  
 Accident site:事故発生現場,Drifting hatches discovered:漂流ハッチ発見



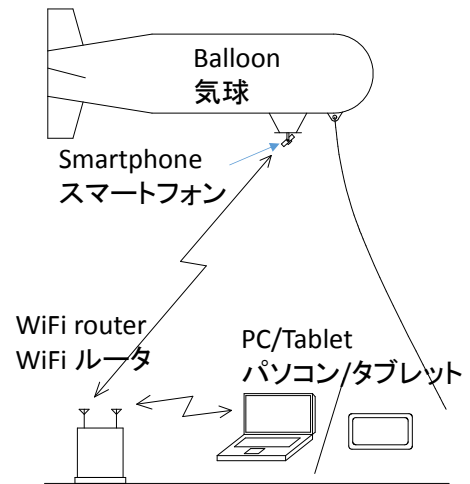


(a) 海岸線への漂着



(b) 浅瀬での座礁

写真-9 BEAGLE3号沈没事故で発生した大型漂流物



(a) システム全体構成



(b) 気球

図-26 気球空撮システム

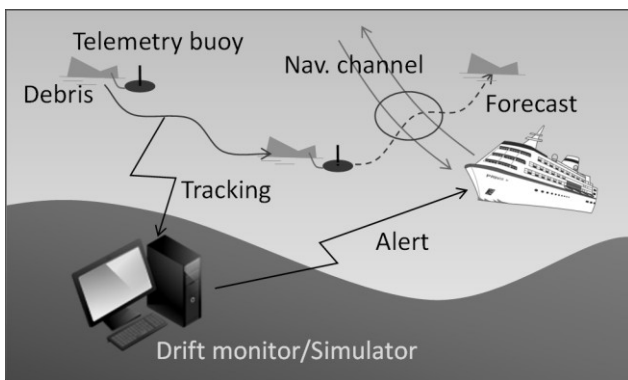


図-25 漂流物へのタグ付と監視

Debris: 漂流物, Telemetry buoy: テレメトリブイ, Tracking: 追跡, Drift monitor/simulator: 漂流監視/予測, Alert: 監視, Forecast: 予測, Nav. channel: 航路

## 4.2 気球空撮システムへの応用

4.1では開発した漂流ブイの比較的直接的な応用例を紹介したが、ここでは組み込み型コンピュータとしての要素的な応用例を考える。具体的には気球空撮システムへの応用を日本作業船協会と共同研究したので、以下で内容を紹介する<sup>15)</sup>。

気球空撮システムは図-26に示す様に、飛行船の形状を持つ気球にカメラを搭載し、リモコンにより空撮を行うシステムである。初期モデルは国土交通省中国地方整備局により開発された。中国地方整備局災害対策用機械に登録されており、東北地方太平洋沖地震の折には宮城県仙台塩釜港において被災状況の空撮を行うなど、人が近づけない危険な場所や離れた場所にある対象物を観察、撮影するには非常に有効な手段である<sup>16)</sup>。その後、日本作業船協会による小型化の検討などが行われていた<sup>17)</sup>。近年はドローンなど無索ヘリコプタによる空撮が関心を集めているが、洋上など、風が比較的強い環境では、有索の気球の方が適している場合も多い。

気球に搭載するペイロードで実現すべき基本機能はカメラの姿勢制御と映像（プレビュー動画）及び画像（静止画）の撮影と伝送である。カメラの姿勢制御は、初期開発モデル及び現行モデルにおいては、ラジコンによるサーボ制御で実現している。映像と画像の撮影に関しては、初期

開発モデルにおいては無線 LAN による制御が可能なコンパクトカメラを用いていたが、その後、当該カメラが廃盤になった為、その後のモデルでは、カメラのビデオ端子から映像信号を取り出し、2.4GHz 帯の特定小電力無線による伝送を行っていた。画像伝送に関してはカメラの SD メモリの代わりに Eye-Fi カードを挿入し無線 LAN 経由で行っていた。この様に機能毎に別々に通信した場合、無線通信での干渉を起こすことも考えられ、無線通信を一つに集約することが望ましい。こうしたことがスマートフォンを活用することで可能となる。更に従来モデルで使用していたコンパクトカメラ自体もスマートフォンに内蔵されたカメラを活用することで不要になるので、構成がシンプルになるとともに装置を小型軽量化することができる。

試作したスマートフォン化気球空撮システムの構成を図-27 に示す。カメラの方位角と俯角の制御は漂流ブイに用いたものと同じ拡張ボードを用いて PWM 信号を発生し、サーボモータを駆動することで実現している。地上局との通信は無線 LAN で一元化し、サーボの制御信号、映像、画像は全て TCP/IP で行う様にしたことにより、従来機に比べてシンプルな構成となった。バッテリーを除く重量では従来機で 485g であったものが、スマートフォン版では 300g となり、38.1%の軽量化が実現できた。200g 弱の軽量化ではあっても、気球での効果は大きい。およそのイメージとしては気球の容積をおおよそドラム缶一本分少なくすることができることになる。

気球空撮システムの、ソフトウェアは、かなりの部分が漂流ブイと共通しているが、漂流ブイにはない機能である画像関係については、プレビュー映像を取得し、リアルタイムで地上端末に転送する UDPpreveiw と撮影した静止画を転送するための FileTransferService の 2 クラスを追加している。地上側の端末用には図-28 に示すようなカメラのプレビュー画面、地図情報画面、気球の制御コンソール画面などを用意した。地上からの操作で写真-10 に示す様な空撮画像を取得することができる。

無人で無線でリアルタイムといった計測ニーズは掘り起こせば、他にも多々あるに違いない。むしろ時代は IoT (Internet of Things) と称して、色々な物に通信デバイスを組み込み、ネットワークで結びつけようとしている。こうした流れの中で、取り敢えずは安価で簡便に機能を実現したい場合、本資料で紹介した技術が解決策を提示してくれる場合がかなりあるのではないかと期待している。

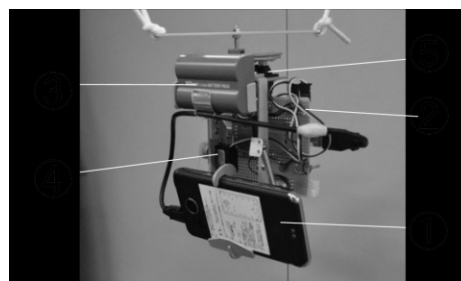
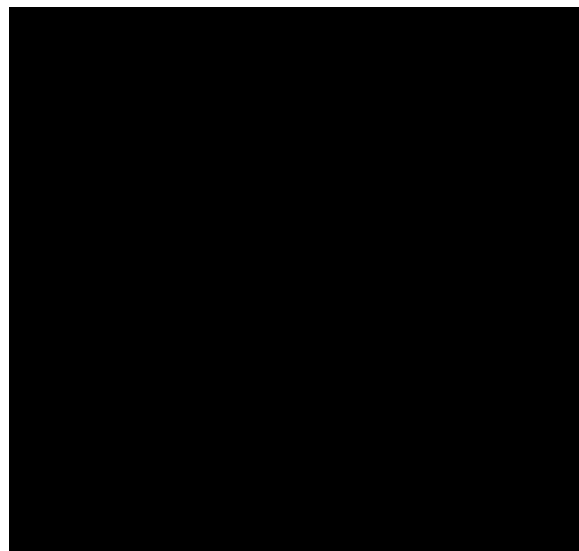


図-27 スマートフォン化システムのハードウェア構成

- ①アンドロイドスマートフォン
- ②USB拡張基板,
- ③バッテリー,
- ④俯角制御サーボ,
- ⑤方位角制御サーボ

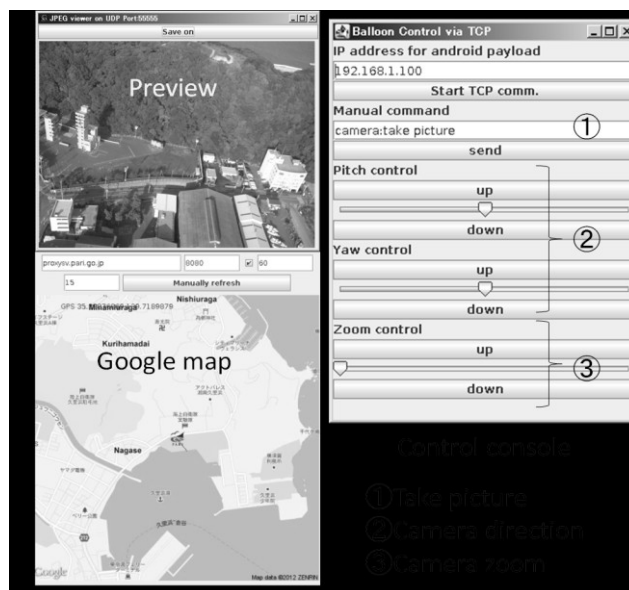


図-28 PC 側ソフトの画面

Preview: プレビュー画面, Google map: マップ表示, ①撮影, ②カメラの向きを制御, ③カメラのズーム



写真-10 スマートフォン化気球空撮システムの画像

## 5. まとめ

海上流出油の漂流捕捉を主目的として簡易漂流テレメトリブイのプロトタイプ開発を行なった。開発においては、アンドロイドOSを搭載したスマートフォンを組み込み機器として活用し、携帯電話のパケット通信を利用することで、簡便且つ安価なブイシステムを構築することができた。実海域における昨日試験を実施し、位置情報に加えて、油の漂流予測に必要な風向風速データをリアルタイムテレメトリにより収集できることを示した。開発したブイの活用として、海上流出油の乱流拡散計測事例と、ブイを用いた油の漂流位置予測方法を検討し、ブイの有用性を明らかにした。さらに大型漂流物のリアルタイム捕捉への応用案と要素技術的な応用展開の可能性として気球空撮システムを検討した。無人で無線でリアルタイムの計測が安価かつ簡便に行える時代となったことは喜ばしい。本資料では海上流出油の捕捉を主題とした開発を紹介したが、こうした技術が他にも色々な用途に活用されていくことを期待したい。その際、本資料が多少なりとも参考になることを願って結びとする。

(2015年5月1日受付)

### 謝辞

本資料で述べた気球空撮システムへの応用の部分は一般社団法人作業船協会との共同研究で実施したもので、その際辰口雅光氏には共同研究のパートナーとして大変お世話になった。また実海域試験並びに2013年度版のブイの製作に関しては、(株)ソニックにご協力頂いた。ここに記して感謝申し上げる。

### 免責事項

本資料で紹介した実海域試験にあたっては、海上保安庁や漁協などの関連機関との調整を事前に行い、監視船を付けるなどの配慮の上で実施しており、本資料で紹介したブイの実海域における使用を無条件に保証してはいない。加えて、本資料で紹介した内容を再利用した結果として生じた損害については一切責任を負わないものとする。

### 参考文献

- 1) 松崎義孝, 藤田勇: 油拡散を考慮した流出油の数値計算法の開発, 港湾空港技術研究所資料, No.1225, 2012.
- 2) 日向博文, 諸星一信, 金津伸好: HFレーダーによる漂流ゴミ回収効率化の試み, 沿岸海洋研究 45(2), 105-114, 2008.
- 3) 加藤直三: 原油流出の影響と対策 ブイによる流出油の追跡, Kanrin, No.35, pp.13-17, 2011.
- 4) M.Fingas: Buoys and Devices for Oil Spill Tracking, *International Oil Spill Conference Proceedings*, Vol. 2011, No. 1, pp. abs9, 2011.
- 5) 藤田勇, 松崎義孝, 白井一洋: スマートフォンを活用した海象情報のリアルタイム収集ブイの試作, 土木学会論文集B3(海洋開発), Vol.69, No.2, I\_1138-I\_1143, 2013.
- 6) 藤田勇, 松崎義孝: 沿岸域向け海洋漂流物捕捉のための簡易漂流テレメトリブイの開発, マリンエンジニアリング学会誌, Vol.49, No.2, pp.11-16, 2014.
- 7) <https://github.com/ytai/ioio/wiki/ioilib-basics>
- 8) 大島浩太, 古谷雅理, 竹本孝弘, 庄司るり: 東京湾・浦賀水道海上における3G電波強度調査, 日本航海学会講演予稿集, Vol.2, No.1, pp.42-45, 2014.
- 9) J.A.Fay: Physical Processes in the Spread of Oil on a Water Surface, *Conference on Prevention and Control of Oil Spills*, pp.463-467, 1971.
- 10) W.J.Lehr, R.J.Fraga, M.S.Belen and H.M.Cekirge, A New Technique to Estimate Initial Spill Size Using a Modified Fay-Type Spreading Formula, *Marine Pollution Bulletin*, Vol.15, pp.326-329, 1984.
- 11) ASCE Task Committee on Modeling of Oil Spills of the Water Resources Engineering Division. State-of-the-Art Review of Modeling Transport and Fate of Oil Spills, *Journal of Hydraulic Engineering*, Vol.122, pp.594-609, 1996.
- 12) 藤田勇, 松崎義孝: 薄い浮遊物の風による運動の過渡応答特性の計測, 土木学会論文集B3(海洋開発), Vol.70,



- No.2, pp.1.804-1.809, 2014.
- 13) 松崎義孝, 藤田勇: 海水面極近傍における水平乱流拡散に関する研究, 土木学会論文集B2(海岸工学), Vol.69, No.2, pp. 1.476-1.480, 2013.
  - 14) M.Fingas: The Basics of Oil Spill Cleanup, 3rd ed., CRC press p.58,2013.
  - 15) 藤田勇: スマートフォンを用いた遠隔リアルタイム計測-海象情報収集漂流ブイ及び気球空撮システムへの応用-, 作業船, No.312, pp.13-18, 2013.
  - 16) 気球空撮システムの活用について～災害対策用機械への新規登録と活用状況～、広島技調じゃけん！通信第15号、2011.
  - 17) 社団法人日本作業船協会:気球を利用した映像情報収集システムの改良、雑誌作業船、 No.300、 pp.27-37, 2010.

港湾空港技術研究所資料 No.1312

2015.9

編集兼発行人 国立研究開発法人 港湾空港技術研究所

発行所 国立研究開発法人 港湾空港技術研究所  
横須賀市長瀬3丁目1番1号  
TEL. 046(844)5040 URL. <http://www.pari.go.jp/>

印刷所 株式会社 シーケン

Copyright © (2015) by PARI

All rights reserved. No part of this book must be reproduced by any means without the written permission of the President of PARI

この資料は、港湾空港技術研究所理事長の承認を得て刊行したものである。したがって、本報告書の全部または一部の転載、複写は港湾空港技術研究所理事長の文書による承認を得ずしてこれを行ってはならない。



古紙配合率70%再生紙を使用しています