ソフトウエアラジオとOpenLayersで作るAIS表示サイト

2018年6月

有限会社サンビットシステム
佐々木伸幸

Sunbit System

SAPPHIRE PRINCESS MMSI:235103357 CS:2HFZ6
那覇クルーズターミナル　2017/6/14

RTL-SDR

RTL2382Uを使ったUSBワンセグチューナが生I/Qデータ をUSBに流せることを利用して、復調をソフトウエアで行う無線受信機。
同調用チップによって受信周波数範囲は変わるが、22MHz - 2200MHzと超広帯域受信機になる。

BSD,Linux, Windowsなどほとんど動作。

https://github.com/osmocom/rtl-sdr

http://osmocom.org/projects/sdr/wiki/rtl-sdr
最も情報が集積されているように見える

Sunbit System

RTL-SDR

Terratec	Black Nooelec	Blue Nooelec	Silver dongle	Nano Nooelec
R820T	R820T	R820T2	R820T2	R820T
no TCXO	no TCXO	TCXO	TCXO	no TCXO

http://livedoor.blogimg.jp/bh5ea20tb/imgs/b/9/b9a45cf4.jpg

AIS=船舶位置自動識別装置

船の位置をVHF帯(162MHz帯2波)で発信

国際的規格
300t以上の客船、500t以上の貨物船に搭載義務
＝プレジャーボート、漁船は積んでない

12W程度の送信電力＝30Kmは到達

Sunbit System

RTL-AIS

RTL-SDRをベースとしてAIS受信に特化したもの
161.075Mhz, 162.025Mhzの2波を信号処理できる

受信したデータをNMEA形式で出力する

https://github.com/dgiardini/rtl-ais

Sunbit System

NMEA形式

```
!AIVDM,2,2,6,B,Bp<Up8888888880,2*32
!AIVDM,1,1,,B,?0474GQa=PEPD003000,2*57
!AIVDM,1,1,,B,16Kdpj@02>:8AitHHbP86nQF0000,0*70
```

データフォーマットの解説
http://catb.org/gpsd/AIVDM.html

type 1,3　　位置情報
type 4　　基地局情報
type 5　　船舶情報

Sunbit System

AISで取れる情報

位置情報

MMSI:	AISで使用される個体識別番号
Lon,Lat:	GPSの緯度経度
Time:	発信時刻
Heading:	船首の向き(360度)
Speed:	速度(ノット)

船舶情報

MMSI:	AISで使用される個体識別番号
Callsign	無線局的なコールサイン（ないことがある）
Name:	船舶名（ないことがある）
ToBow	GPS位置から船首までの長さ
ToStern	GPS位置から船尾までの長さ
ToPort	GPS位置から左舷の長さ
ToStarBoard	GPS位置から右舷の長さ

Sunbit System

gpsd

gpsの情報を解析、中継するソフトウエア
AISのNMEAデコード機能を持っている

gpsをソースとして位置情報サーバになったり、他からのデータを中継するサーバになったりできる

http://www.catb.org/gpsd/

Sunbit System

AISデータの受信頻度と格納方法

TYPE 1,3	位置情報	数秒 - 十数秒に一度(停泊等は十数分)
TYPE 4	基地局	十数分に1度
TYPE 5	船舶情報	十数分に1度

位置情報は蓄積しないと航跡をたどる等ができない
　位置情報だけでは船舶の詳しい情報はわからない
　位置情報にある発信時刻はUTC時刻とあるが秒だけ

船舶情報は同じ船舶なら最新だけあればよい
（基地局も同じ）

Sunbit System

PostgreSQL+PostGISに格納する

AIS情報の緯度経度をPostGISのgeom形式で格納
測地系に関する演算機能もある

送られてくるGPSの緯度経度
　測地系WGS84を600000倍したもの

geom形式
位置情報を点・線・面等に分類して格納
EPSG:4326 = WGS84

gpsd2pgsqlを作成
　受信データ/600000をEPSG:4326で格納

Sunbit System

gpsd2pgsql.pl

use Net::GPSD3;

gpsdにjsonモードでつなぐ
DBあける

LOOP: 終了まで(SIGTERMとか)
　　jsonデータを解析してタイプから動作を判断

　　case 位置情報:
　　 INSERT MMSI,位置,格納時刻(位置は生値とgeometryで格納)

　　case 船舶or基地局:
　　 UPDATE MMSI,付属情報　　（最初だけINSERT)
END LOOP

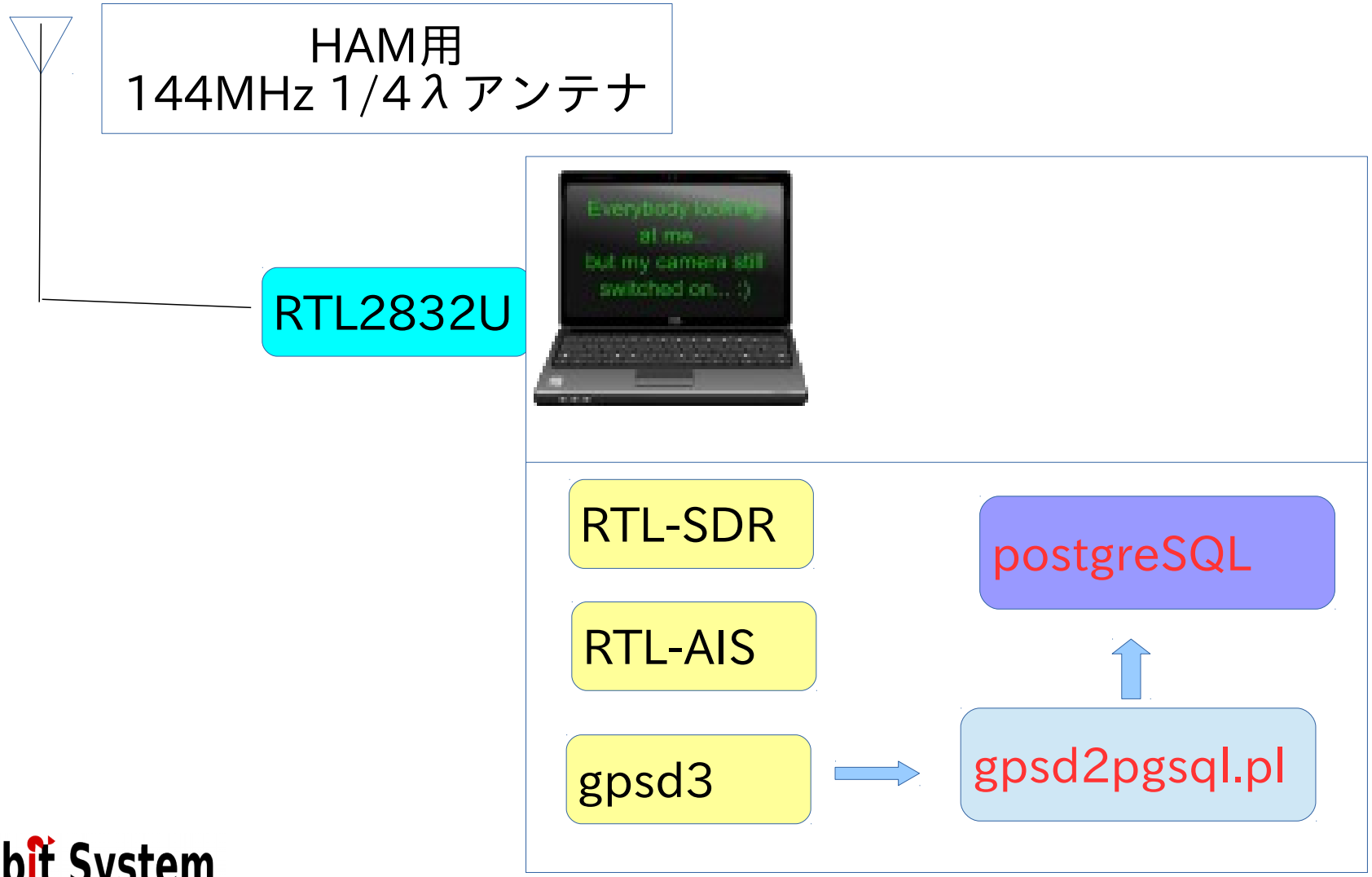
Sunbit System

DBテーブル

```
position (
    char(10) mmsi;          # AIS識別子
    numeric lat,lat;        # 緯度経度
    geometry geom;          # 緯度経度を空間座標系にしたもの
    timestamp  createat  # 格納時刻
);

vessel (
    char(10) mmsi;          # AIS識別子
    text name;              # 船舶名
    int type;               # 貨物船、客船などの種別
    int to_bow,to_stern, to_port, to_starboard;  # GPS位置
);
```

Sunbit System

接続図



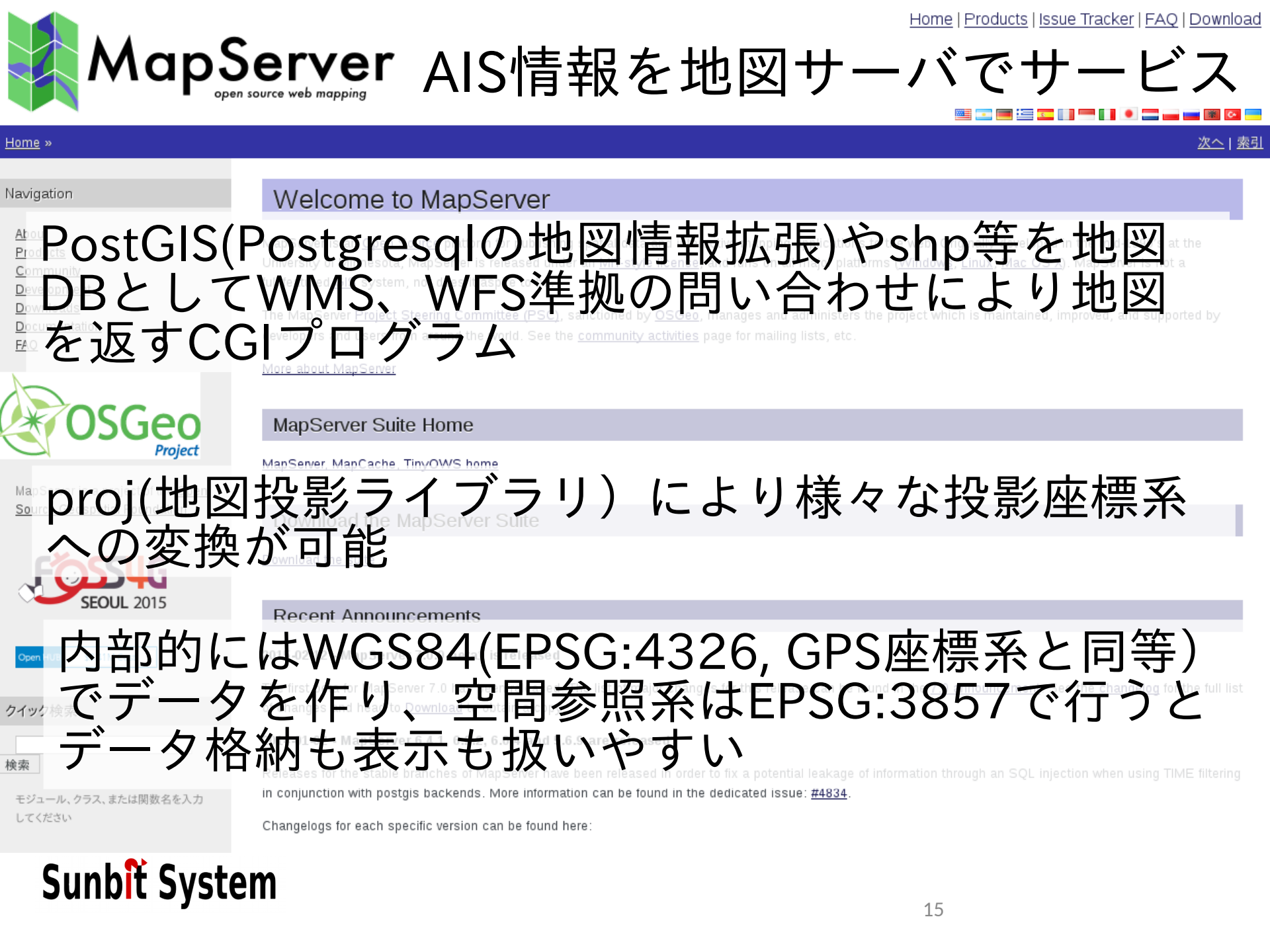
Sunbit System

位置情報と船舶情報の加算

位置情報と船舶情報をJOINして取り出す必要
mmsiをキーにVIEWを作る

```
CREATE  VIEW vessel_view AS
 SELECT DISTINCT "position".id, "position".type, "position".status,
    "position".mmsi,
    st_transform("position".geom, 4326)::geometry(Geometry,4326) AS
geom,
    "position".lat, "position".lon, "position".course, "position".speed,
    "position".heading, "position".turn, "position".second,
    vessel.shipname,
    vessel.shiptype, vessel.imo, vessel.callsign, vessel.to_bow,
    vessel.to_stern, vessel.to_port, vessel.to_starboard, vessel.draught,
    vessel.destination, "position".create_at,
    date_part('epoch'::text, "position".create_at) AS created
   FROM "position"
   JOIN vessel ON "position".mmsi::text = vessel.mmsi::text;
```

Sunbit System



AIS情報を地図サーバでサービス

PostGIS(Postgresqlの地図情報拡張)やshp等を地図DBとしてWMS、WFS準拠の問い合わせにより地図を返すCGIプログラム

proj(地図投影ライブラリ）により様々な投影座標系への変換が可能

内部的にはWGS84(EPSG:4326, GPS座標系と同等）でデータを作り、空間参照系はEPSG:3857で行うとデータ格納も表示も扱いやすい

Sunbit System

AIS情報を地図サーバでサービス

MapServerの定義にWFSとしてAIS情報を追加

```
  LAYER
    NAME 'vessel_pos'
    type POINT
    METADATA
      'wfs_featureid'       'vessel_pos'
      'wfs_enable_request'  '*'
      "wfs_getfeature_formatlist" "*,geojson"
      'gml_featureid'       'vessel_pos'
      'gml_include_items'   'all'
      'ows_title'       'vessel_pos'
      'wfs_srs'         'EPSG:4326 EPSG:3857'
    END
    CONNECTION "user=wms host=192.168.1.1 dbname=ais port=5432"
    CONNECTIONtype postgis
    DATA "geom from (select geom,id,to_number(mmsi,'999999999') as
mmsi,status,shipname,shiptype,callsign,lat,lon,(360 - heading) as heading,speed,(360 -
course) as course,turn,to_bow,to_stern,to_port,to_starboard,(to_bow + to_stern) as
length,(to_port + to_starboard) as width,draught,destination,create_at,created from
vessel_view) as foo using unique id using srid = 4326"
```

Sunbit System

AIS情報を地図サーバでサービス

MapServerに問い合わせるとgeojsonでデータが返る

GET /cgi-bin/mapserv?
service=WFS&request=GetFeature&version=1.1.0&typename=vessel_pos&srsname=EPSG:4326&OUTPUTFORMAT=geojson&....

{
"type": "FeatureCollection",

"features": [
{ "type": "Feature", "properties": { "id": "10032950", "mmsi": "431003094", "status": "0", "shipname": "AKEBONO MARU", "shiptype": "83", "callsign": "JD3204", "lat": "42.594772", "lon": "141.630180", "heading": "269.0", "speed": "21.5", "course": "286.0", "turn": "127", "to_bow": "75", "to_stern": "24", "to_port": "16", "to_starboard": "1", "length": "99", "width": "17", "draught": "4.6", "destination": "JP HKP OFF", "create_at": "2018-06-06 18:49:27", "created": "1528310967" }, "geometry": { "type": "Point", "coordinates": [141.63018, 42.594772] } }
]

Sunbit System

AIS情報を地図サーバでサービス

気をつけること

WFSの時刻検索は数値比較でないとうまくいかない
(UNIX Epoc等。 YYYY-MM-DD形式はダメ)
>viewにUNIX Epocを追加しておくなど

Sunbit System

OpenLayersで船舶を描画する

OpenLayers
https://openlayers.org/

OpenLayersはブラウザで地図データを表示する、
JavaScriptで組まれたオープンソースライブラリ
2条項 BSDライセンスで提供されている。

Sunbit System

OpenLayersの基本

平面直角座標系(投影座標)や初期表示範囲を決める

ベースレイヤを作成する　(OSMでいいよね)

重ねるレイヤを作成する　(船舶情報)

Mapオブジェクトにレイヤを追加する

レンダリング

Sunbit System

空間座標系や初期表示範囲

データはEPSG:4326, 投影座標はEPSG:3857
中心緯度経度は表示したいところの真ん中
初期表示範囲はズーム値で14くらい

座標系変換

```
view = new ol.View({
          center: ol.proj.transform(
                [141.6263,42.6337],
                'EPSG:4326','EPSG:3857'),
          zoom: 14
        });
```

Sunbit System

ベースレイヤを作成する

OpenStreetMapを使うなら簡単

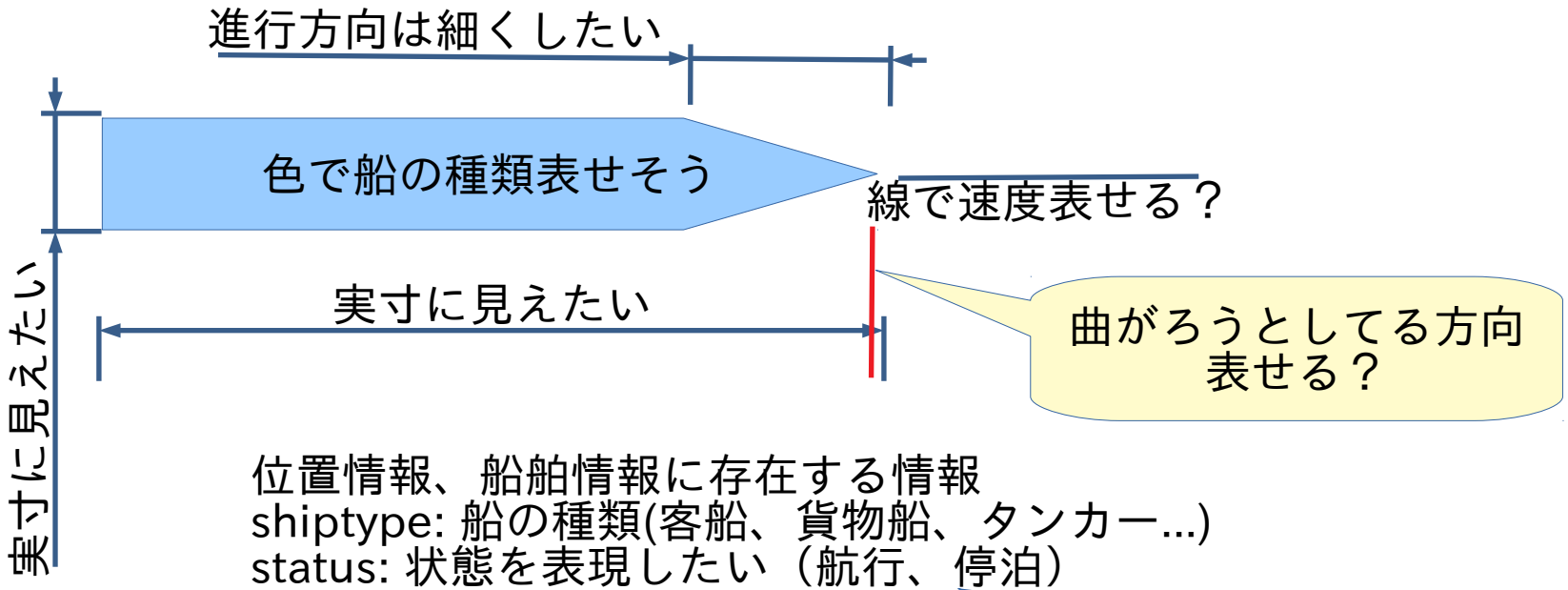
```
// ベースレイヤを作成
lbase = new ol.layer.Tile({
            source: new ol.source.OSM()
        });
```

タイル型レイヤに
ソースとしてOSMを指定するだけ

Sunbit System

重ねるレイヤを作成する　(船舶情報)

船舶情報をどう表現する？



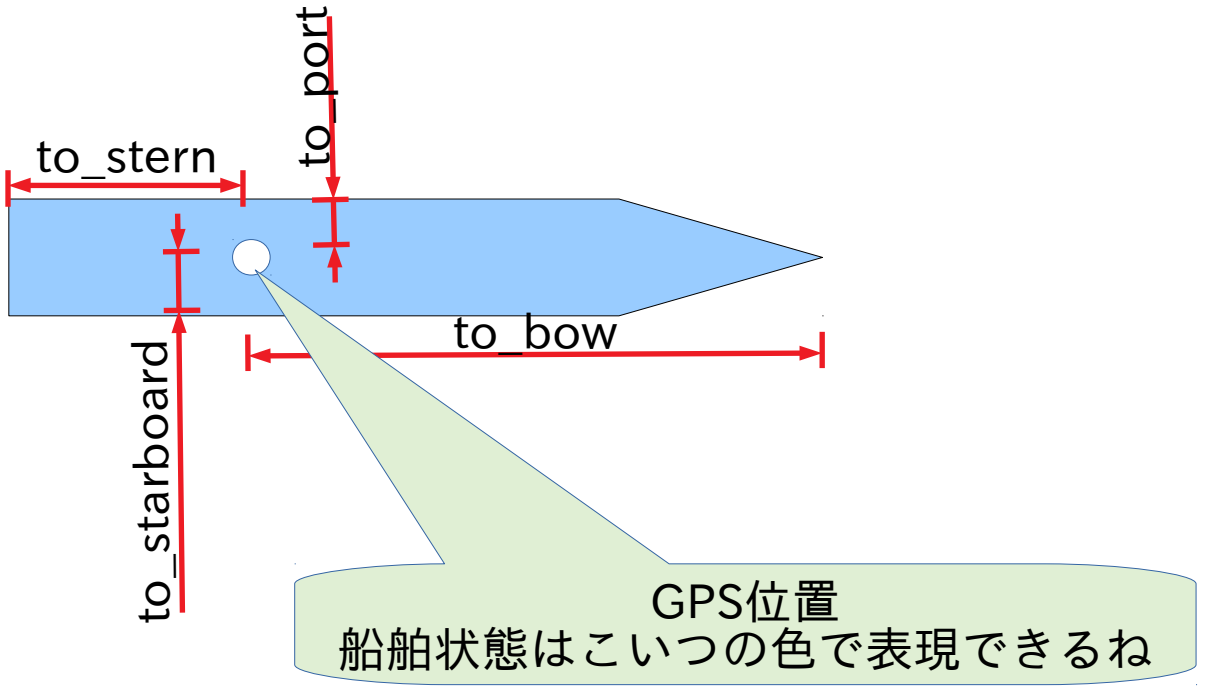
位置情報、船舶情報に存在する情報
shiptype: 船の種類(客船、貨物船、タンカー...)
status: 状態を表現したい（航行、停泊）
speed: 速度を表現したい
turn:　曲がろうとしてるかを知りたい

状態どうしよう？

Sunbit System

重ねるレイヤを作成する　(船舶情報)

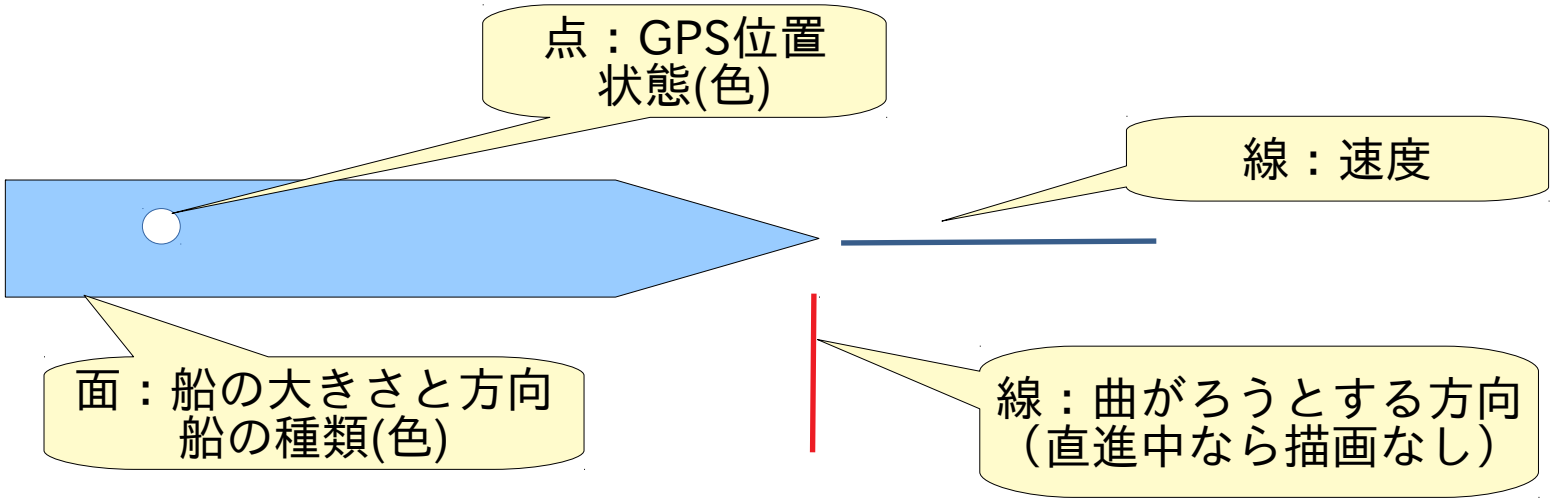
GPS位置情報は点情報　船はでかい
　to_bow, to_starboard てそういうことか！



Sunbit System

重ねるレイヤを作成する　(船舶情報)

必要なレイヤ



位置情報、船舶情報に存在する情報
shiptype: 船の種類(客船、貨物船、タンカー...)
status: 状態を表現したい（航行、停泊）
speed: 速度を表現したい
turn:　曲がろうとしてるかを知りたい

Sunbit System

重ねるレイヤを作成する　(実際の長さ比率)

縮尺係数

平面直角座標系は原点位置からの距離
　http://www.gsi.go.jp/sokuchikijun/datum-main.html
原点から遠い程、同じ距離は地図上で長く投影される

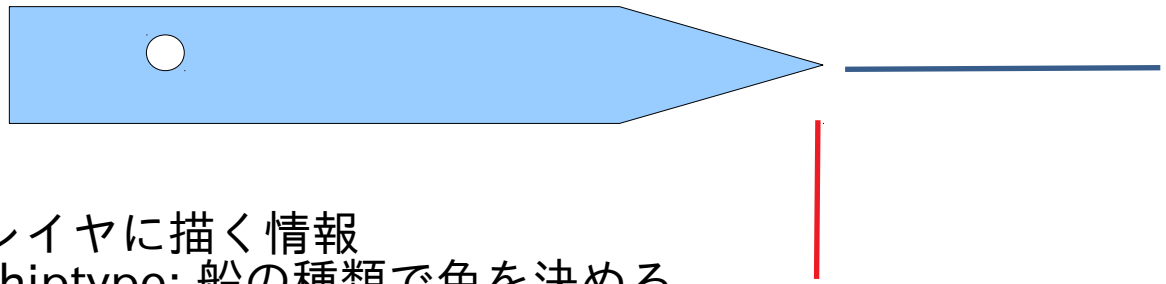
Openlayersには縮尺係数を得る関数がある

```
e = getExpansion([緯度, 経度]); //位置での係数
bow = to_bow * e;  // 平面上での地図上長さ
```

Sunbit System

重ねるレイヤを作成する　(船舶情報)

1) GPS位置に点を描き、状態色で塗る
2) 船舶の基準形に全長全幅を乗じて形を作り
   GPS位置からto_port, to_stern分ずらして配置し、
   船舶の方向に傾けて描く。船舶種別色で塗る
3) 船舶の先端位置から速度分の線を進行方向に描く
4) 旋回中であれば速度線と同じように線を描き、
   旋回方向に回転する

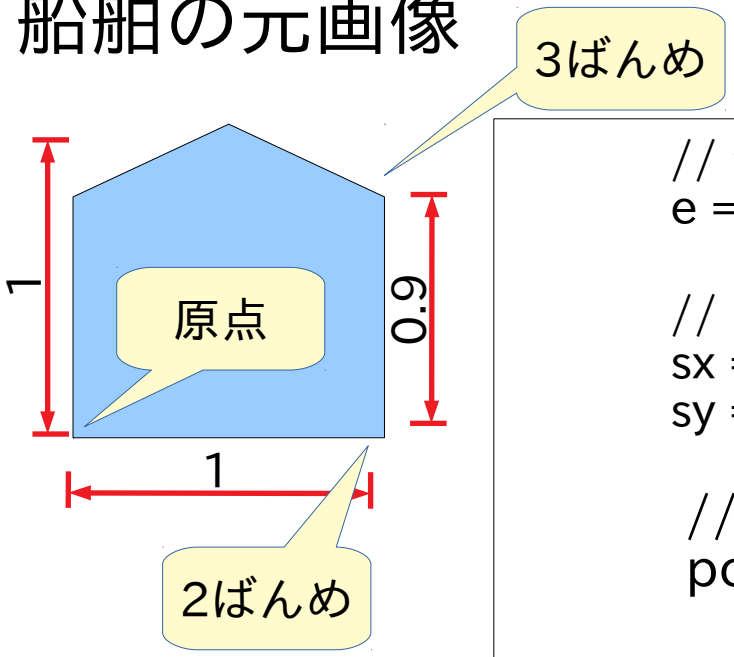


レイヤに描く情報
shiptype: 船の種類で色を決める
status: 船の状態で色を決める
speed: 速度と長さを比例関係で長さを決める
turn:　長さは一定で旋回方向に回転する

Sunbit System

船舶のかきかた

船舶の元画像

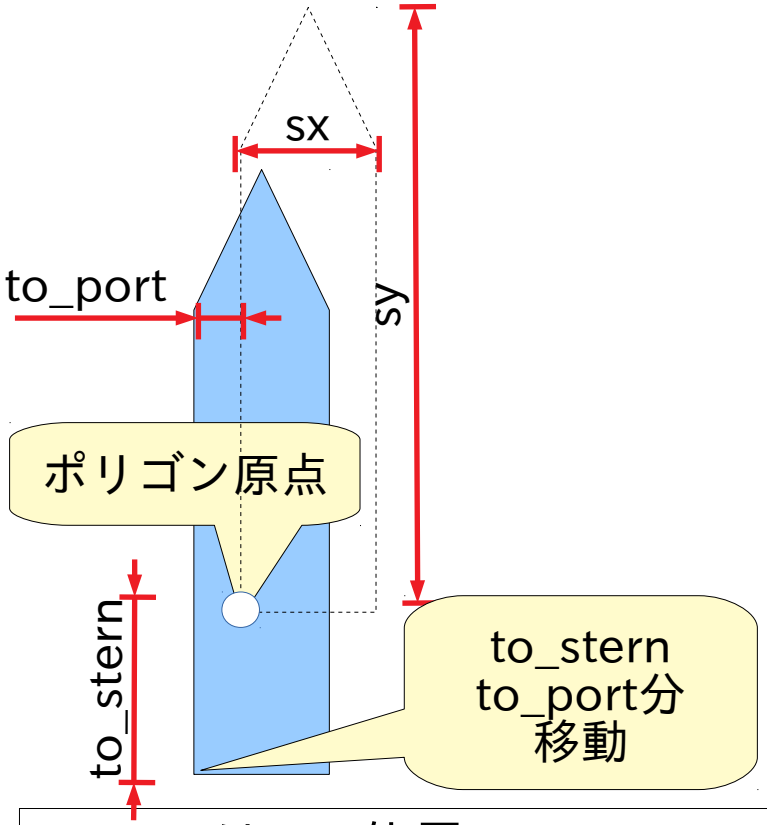


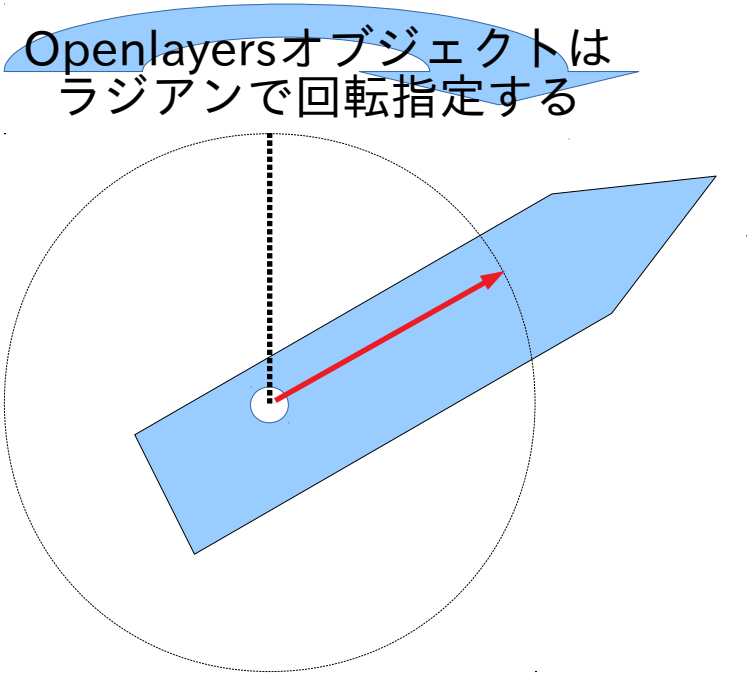
実寸値でpolygon作成

```
// 縮尺係数
e = getExpansion(g.getCoordinates());

// 全長(x), 全幅(y)
sx = (to_port+to_starboard) * e;
sy = (to_bow+to_stern) * e;

 // 船舶の形を作成
 poly = new ol.geom.Polygon([[
        [sx*0, sy*0],              // 原点
        [sx*1, sy*0],              // 2ばんめ
        [sx*1, sy*0.90],           // 3ばんめ
        [sx*0.5, sy*1],
        [sx*0, sy*0.90],
    ]]);
```

Sunbit System

船舶のかきかた






```
// cx,cyはGPS位置

poly.translate(cx-(to_port*e),
                    cy-(to_stern*e));
```

```
// AISは度で来る
angle = p.heading * (Math.PI / 180);

// GPS位置を中心に回転
poly.rotate(angle,[cx,cy]);
```

Sunbit System

地物のレイヤ

```
// ベクタレイヤのソースはMapServerへの問い合わせ
svessel = new ol.source.Vector({format: new ol.format.GeoJSON(),
        url: function(extent){ return 'http://wms.3bit.co.jp/cgi-bin/mapserv?'...

function createStatus( );    // GPS位置と船舶状態のポリゴン作成
function createCourse( );  // 速度線を作成
function createTurn( );      // 転回方向線を作成
function createVessel( );   // 船舶のポリゴンを作成

// 同じデータソースを元にそれぞれの船舶情報のレイヤを作る
lstatus = new ol.layer.Vector({source: svessel, style: createStatus});

lcourse = new ol.layer.Vector({source: svessel,opacity: 0.7,
                                style: createCourse});

lturn = new ol.layer.Vector({source: svessel, opacity: 0.7,
                                style: createTurn });

lvessel = new ol.layer.Vector({source: svessel,opacity: 0.7,
                                style: createVessel});
```

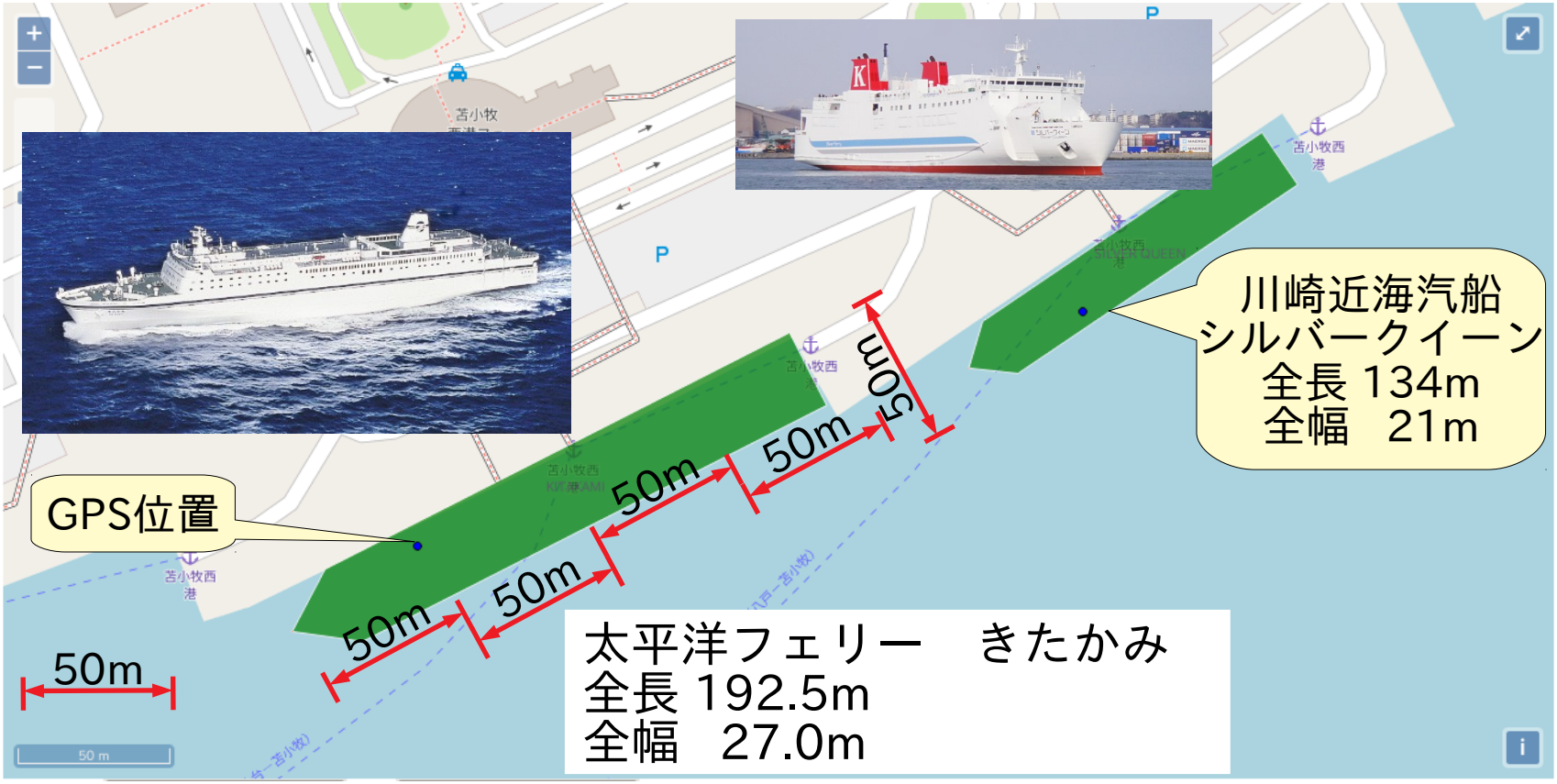
Sunbit System

レイヤ合体

```
// スクロールバー、拡縮などの操作部品
controls = new ol.control.defaults().extend([
                new ol.control.MousePosition({projection: 'EPSG:4326',
                        coordinateFormat: ol.coordinate.createStringXY(4)}),
                new ol.control.ScaleLine(),
                new ol.control.ZoomSlider(),
                new ol.control.FullScreen()]);

// マップ生成
map = new ol.Map(
            {target: 'map',                                          //DOMのmapに描画
             layers: [lbase,lvessel,lcourse,lturn,lstatus],  // レイヤ描画は左が下
             view: view,                                             // 先に作った表示位置
             controls: controls                                      // ブラウザの操作部品
            });
```

Sunbit System

表示結果



http://www.taiheiyo-ferry.co.jp/senpaku/img/pic_kitakami2.jpg
http://www.silverferry.jp/images/pages/shipGuide_queenPhotoMain.jpg

Sunbit System

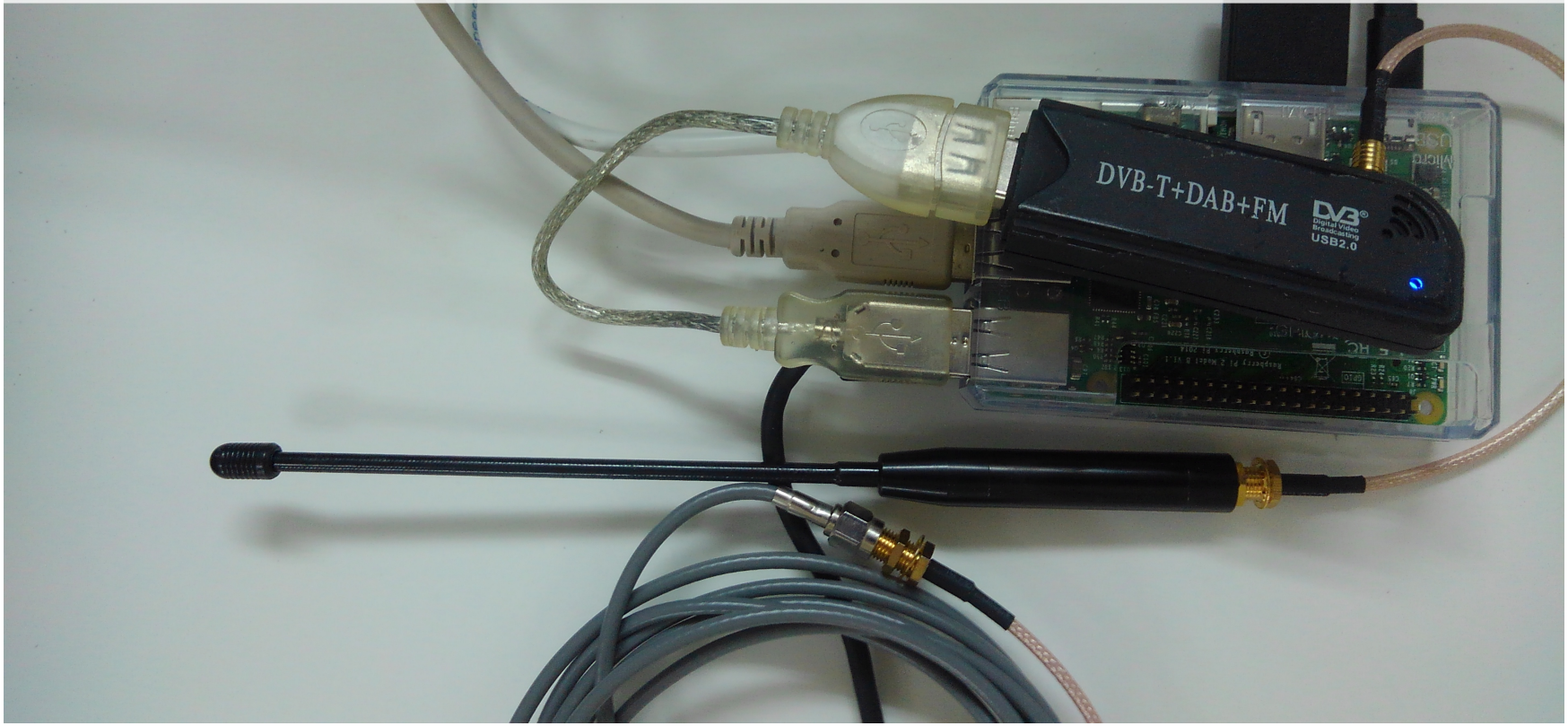
gpsdais2pgsql

ここで作成した定義ファイル、スクリプト等をgpsdais2pgsqlとして公開しました。

http://ossdesk.3bit.co.jp/?action=cabinet_action_main_download&block_id=104&room_id=1&cabinet_id=6&file_id=29&upload_id=61

おまけ

Raspberry PI 2でAIS受信装置作りました。
（openSUSE arm7にrtl_ais載せただけです）



Sunbit System

Sunbit System 有限会社サンビットシステム

- 1998年8月25日　札幌市東区に登記
- 2009年より札幌市豊平区平岸
- ITインフラ構築、ITシステム設計支援
- ソフトウエア受託開発,パッケージ開発
- IT教育など