サケの回遊行動自動追跡システムの開発
―中間報告―

鎌　田　清　春

1．はじめに

サケは産卵のために生まれた川に回帰する現象は古くから知られているが、そのメカニズムは未だ多くの謎を残しており、これまで、1.太陽コンパス説、2.偏光説、3.星座コンパス説、4.磁気コンパス説、5.海流説、6.母川の匂い記憶説などが母川回帰機構にかかわっているのではないかと言われているが、どの説も実証はされていない。

北海道大学においては水産、理、薬学部の研究者がその研究に取り組んでおり、いくつかの点で明らかになりつつある。

研究方法の一つに、上記 1～ 6の行動に係わる器官に施術し、小型の超音波発振器（ピンガータグ）を装着した後放流し、受波機をとりつけた船で追跡する方法が取られており、回帰機構を調べる上で不可欠の実験である。

降海型サケを用いて実験を行うためには膨大な費用と人員が必要であり、現実には実行は難しい。そこで、陸封型のベニサケでサケと同様の行動形態をとるヒメマス（チップ）を用い、北海道・洞爺湖で回遊実態の調査・研究が進められている。ヒメマスの回遊行動追跡の実験を行ううえで以下のような問題点がある。

1. 1個の受波機で追跡可能な個体は1匹であり、又、1日に1匹の追跡が限度である。
2. 数日間の追跡が必要であるが体制および人的な労力（かなりの重労働）から言っても現状では　無理である（現在は1日12～14時間）。
3. 多数のヒメマスの行動を追跡するためには何度も同じことを繰り返さなければならず、同一条件のもとで複数のメカニズム調査が出来ない。
4. 湖上の正確な位置を特定し記録するのが大変困難である。
5. 実験者の危険の回避（特に夜間）。

これらの点を解決するための自動追跡システムが求められており、その開発を目的としている。

なお、本研究は平成6年度科学研究費補助金（奨励研究（B））の支援のもとに行われた。

1.1 システムとして開発すべき内容

1) ロボット船体の設計および推進方法の確立
2) 超音波の到来方向の自動推定
3) 湖上における船の位置の特定
4) 個体の水深および水温等の情報取得
5) 各種情報を無線通信により陸上に電送するための送受信装置
6) 情報の記録・解析・図示するためのプログラム
7) ロボット船のコントロールおよび情報制御用マイコンの周辺装置・ソフト

1.2 これまでに開発した内容

1. 船の位置の特定

イ、湖上における船の位置をＧＰＳ(Global Positioning System) を用いて特定する方法を用いた。ＧＰＳ受信機は、ソニー製IPS-3000を用いた。

ロ、ＧＰＳ受信機の出力データをパソコンに取り入れ記録するモデムとソフトの開発を

行った。

（開発したソフトは３に示す）

ハ、ＧＰＳ受信機の誤差補正をＤＧＰＳ(Difrencial G.P.S)で行うことを検討し実験した。実験の結果、他の誤差要素との関係から必要がないことが判明した。

2. 情報の記録・解析・図示するためのプログラムの開発を行った。

取得したＧＰＳの位置情報をパソコンに記録するとともにパソコン上の洞爺湖（支笏湖）地図上にトレースするためのソフトの開発を行った。

3. ロボット船の情報を無線通信によって湖岸への送信することが可能である事を確認した。現在水産学部に許可されている169.65MHz FM　１Ｗ　の移動局の使用が可能であるかどうかを 144 MHz のアマチュア無線機を用 い、洞爺湖一円の通信テストを行い充分使用が可能であることを確認した。

２．ＧＰＳ受信機関係

2.1 ＧＰＳ受信機の概要

ＧＰＳ(Global Positioning System) は海上、陸上を問わず、航空機、ロケットなど、地球上のほぼ全地域、全天候下で連続的に単独測位可能な衛星航法システムであり、同時に移動体の速度や進行方向、正確な時刻も求めることが出来る。（ＧＰＳは米国国防総省で開発されたものであり、いくつかの問題点もある）

使用したＧＰＳ受信機（ｿﾆｰ IPS-3000）の仕様は以下に示す。

〔電気的仕様〕

受信周波数	：	1575.42 MHz（L1帯　C/Aコード）
受信感度	：	－130dB 以下
電源電圧	：	＋5V　　（絶対最大定格電圧　＋7V）
消費電流	：	190 mA （ 標準値）

〔性能仕様〕

受信チャンネル数	：	８チャンネル
（精度）位置精度	：	９ｍ（ＤＯＰ　４、S/A　Off 時の標準値）
速度精度	：	0.56km/h（ＤＯＰ　４、S/A　Off 時の標準値）
高度精度	：	25m（ＤＯＰ　４、S/A　Off 時の標準値）
最大速度	：	500km/h
アップデートレート	：	１秒毎
初期測位時間	：	１分（アルマナックデータがある場合の標準値）

〔環境仕様〕

動作温度	：	－２０℃～＋７０℃
保存温度	：	－４０℃～＋８５℃
防水性能	：	防噴流型（ＪＩＳ５級）

〔インターフェース仕様〕

通信方式	：	非同期全２重方式
伝送速度	：	９６００bps
データ長	：	８ビット
ストップビット長	：	１ビット
パリティ	：	なし
電気レベル	：	ＴＴＬレベル
入出力コード	：	ＡＳＣＩＩコード

2.2 ＧＰＳ受信機より送出されるメッセージ（例）

```
123456789*123456789*123456789*123456789*123456789*123456789
SONY739602154003050N4304226E14120482+006100026796021540030 49
E3BIFFFXWGoABbCLABUFgCKEGPABGBEFGTDrACADfFNaDJEE                ┐
```

A	: SONY73	ソフトウェアのバージョン
B	: 9602154003050	年月日、曜日、時分秒（ＵＴＣ）
C	: N4304226	緯度（北緯、度、分、秒）
D	: E14120482	経度（東経、度、分、秒）
E	: +0061	ＧＰＳ高度（ｍ、ＷＧＳ－８４測地系）
F	: 000	速度（ｋｍ／ｈ）
G	: 267	進行方位（度）
H	: 9602154003049	３～７の計算時刻
IJK	: E3B	ＤＯＰ値、測地計算モード、位置データ
L	: IFFFX	｝
M	: WGoAB	
N	: bCLAB	
O	: UFgCK	｝ １～８衛星の状態
P	: EGPAB	
Q	: GBEFG	
R	: TDrAC	
S	: ADfFN	｝
TUVVW	: aDJEE	その他（ユーザーに直接関係しない情報）

2.3 ＧＰＳ情報表示とデータ収集画面

***　ＧＰＳ情報表示　***

現在時刻（JST）　1996/02/15　09:35:13

現在位置

緯度…　N 43°04'24"
経度…　E141°20'45"

高度…　+0110m
速度…　007Km/h
方向…　169°

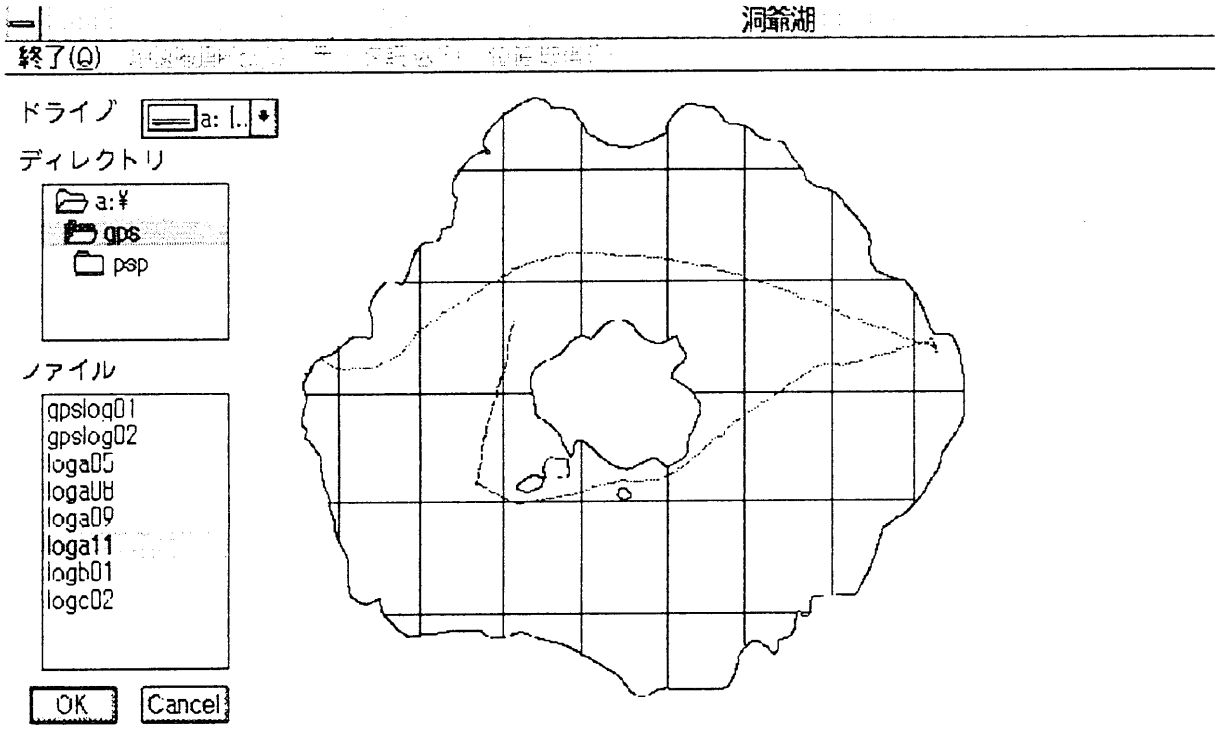
計算モード
4 衛星測位

衛星情報

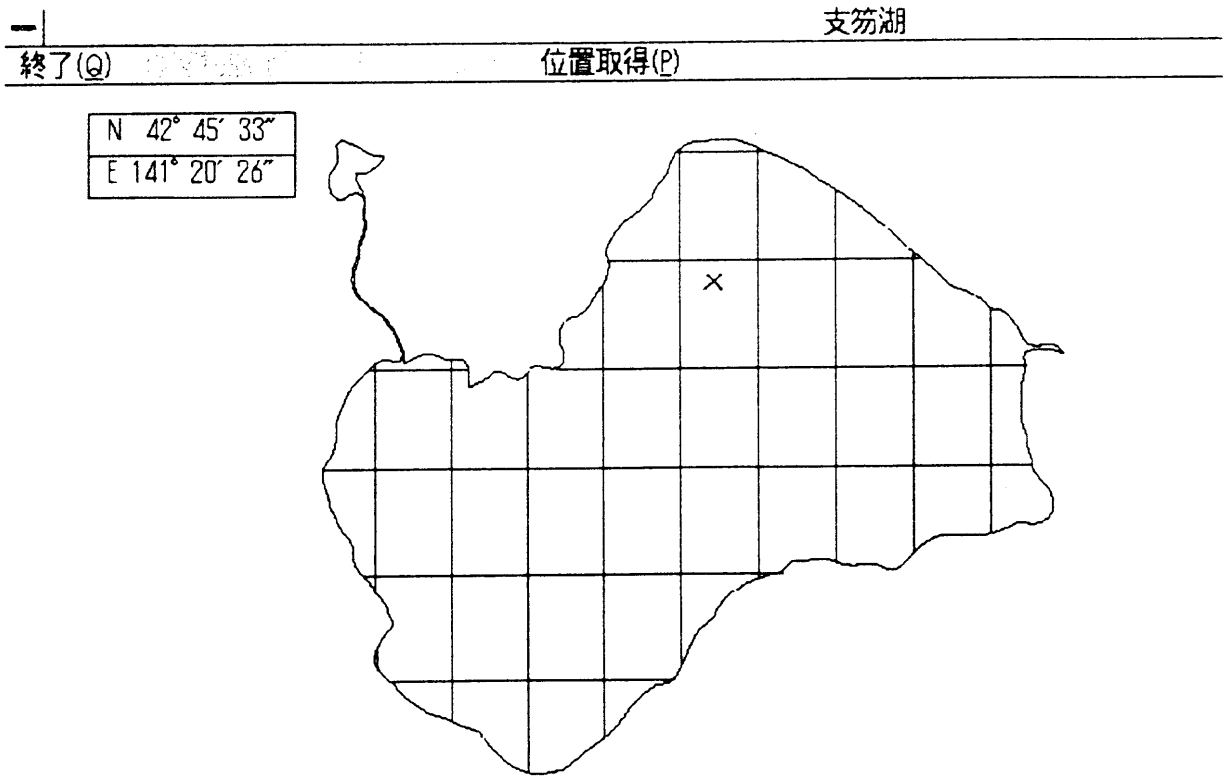
ﾁｬﾝﾈﾙ	衛星	状態	仰角	方向
CH1	SV- 9	F	+50	+50
CH2	SV-23	A	+60	-150
CH3	SV-26	A	+20	+110
CH4	SV-21	F	+50	-80
CH5	SV- 5	A	+60	+140
CH6	SV- 7	F	+10	+40
CH7	SV-20	A	+40	+180
CH8	SV- 1	F	+30	-60

ESC ： 終了　　F10 ： 記録開始

2.4 洞爺湖地図及び軌跡取得表示画面（例）



2.4 支笏湖地図及び位置取得表示画面（例）



３．ＧＰＳ情報にもとずく処理プログラム（ＭＳ－Ｃを使用）

```
/*********************************************************
        ＧＰＳ情報表示  GPS.C
*********************************************************/
#include <malloc.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <io.h>
#include <signal.h>
#include <dos.h>
#include "gps.h"
#include "gpslib.h"
#include "displib.h"
#include "sio.h"
#include "glib.h"

int     SIO_SPEED;

void _interrupt vect6( void )
{
}

/*********************************************************/
void DataDisp( receive_t *rb )
{
        int     ret,flg,lgflg;
         int    i,count;
        char  *pathname,*s;
       static uchar *ch[] = {
                "@CH1,2,3,4,5,6,7,8",
                "@CH9,10,11,12,13,14,15,16",
                "@CH17,18,19,20,21,22,23,24",
                "@CH25,26,27,28,29,30,31,32",
                NULL
        };
        FILE    *fp;

        ginit(); screen(2,0,0,3); cls(2);
        vramfill( ' ',WHITE );
        InfoDisp();
        flg = 0;
        rxempty();
        lgflg = 0;
        count = 0;
        for (;;) {
                ret = ReceiveCR( rb->rbuff );
                if ( ret == -1 ) {              /* ESC key */
                        if ( flg == 0 ) {
                                fclose( fp );
                                break;
                        } else {
                                if ( flg == 1 ) MonClose();
                                        else    MapClose();
                                flg = 0;
                        }
                }
                if ( ret == -2 && flg == 0 ) {          /* F10 key */
                        if ( lgflg == ON ) {
                                fclose( fp );
                                vrams( 49,20," F10 : 記録開始 ",R_GREEN );
                                lgflg = OFF;
                        } else {
                                for ( i=1; i<100; i++ ) {
                                        sprintf( pathname,"gpslog%02d",i );
                                        if ( access( pathname,0 ) ) {
                                                fp = fopen( pathname,"wb" );
                                                break;
                                        }
                                }
                                vrams( 49,20," F10 : 記録終了 ",R_GREEN );
                                sprintf( s,"%d",count );
```

```
                        vrams( 66,20,s,GREEN );
                        lgflg = ON;
                }
        }
        if ( ret == -3 ) {              /* F6 key */
                if ( flg == 2 ) MapClose();
                if ( flg == 1 ) {
                        MonClose();
                        flg = 0;
                } else {
                        MonOpen();
                        flg = 1;
                }
        }
        if ( ret == -4 ) {              /* F1 key */
                if ( flg == 1 ) MonClose();
                if ( flg == 2 ) {
                        MapClose();
                        flg = 0;
                } else {
                        MapOpen();
                        flg = 2;
                }
        }
        if ( ret < 0 ) continue;
        if ( RxJudge( rb ) != VALID ) continue;
        if ( flg == 1 ) {
                MonDisp( rb );
                continue;
        }
        if ( flg == 2 ) {
                MapDisp( rb );
                continue;
        }
        DispTime( rb );
        DispJyoukyou( rb );
        DispEiseiInfo( rb );
        DispNew( rb );
        if ( lgflg == ON & (rb->ss % 10) == 0 ) {
                count++;
                if ( count>7200 ) {
                        fclose( fp );
                        vrams( 54,20," F10 : 記録開始 ",R_GREEN );
                        lgflg = OFF;
                        continue;
                }
                fputs( rb->rbuff,fp );
                sprintf( s,"%d",count );
                vrams( 66,20,s,GREEN );
        }
    }
    vramfill( ' ',WHITE );
}

/*********************************************************/
void main()
{
    uchar           *vramsave,*s;
    int             x,y;
    receive_t       *rb;

    signal( SIGINT,SIG_IGN );
    SIO_SPEED = 7;
    cursor_position( &x,&y );
    winit();
    sio_init( SIO_SPEED,BIT8,PN,S1O );
    tm_open();
    if( (s = malloc( sizeof(receive_t) ) ) == NULL ){
            puts( "メモリが確保できません. " );
            return;
    }
    rb = (receive_t *)s;
    if( (vramsave = malloc(80*25*3) ) == NULL ){
            puts( "メモリが確保できません. " );
```

```
                return;
        }
        get_box( 0,0,80,25,vramsave );
        vramfill( ' ',BLUE );
        cursor_sw( OFF );
        VECTSAVE6 = _dos_getvect( 6 );
        _dos_setvect( 6,vect6 );
        rts( ON );
        DataDisp(rb);                   /* display */
        rts( OFF );
        cursor_sw( ON );
        put_box( 0,0,80,25,vramsave );
        free(vramsave);
        free(s);
        wend();
        sio_iend();
        tm_close();
        locate( x,y );
        _dos_setvect( 6,VECTSAVE6 );
}


/***** GPSDSP.C *****/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include "gps.h"
#include "gpslib.h"
#include "displib.h"

/*************************************************************
        GPS表示の画面
*************************************************************/
static uchar *info[] = {
"                    ***  ＧＰＳ情報表示  ***                    ",
"                                                                ",
"  ┌──────────────────────────────────────────────┐  ",
"  │  現在時刻 (JST)                                                │  ",
"  ├──────────────┬──────────────────────────┤  ",
"  │  現在位置                │  衛星情報                                │  ",
"  │                          │  ﾁｬﾝﾈﾙ 衛星    状態   仰角    方向  │  ",
"  │  緯度…  N   °  '  ¥″  ├──────────────────────┤  ",
"  │  経度…  E   °  '  ¥″  │   CH1   SV-                              │  ",
"  │                          │   CH2   SV-                              │  ",
"  │  高度…        m          │   CH3   SV-                              │  ",
"  │  速度…        Km/h       │   CH4   SV-                              │  ",
"  │  方向…        °          │   CH5   SV-                              │  ",
"  ├──────────────┤   CH6   SV-                              │  ",
"  │  計算モード              │   CH7   SV-                              │  ",
"  │                          │   CH8   SV-                              │  ",
"  └──────────────┴──────────────────────────┘  ",
NULL
};

void InfoDisp( void )
{
        int     y;

        vramfill( ' ',WHITE );
        y = 0;
        while ( info[y] != NULL ) {
                vrams( 7,y+2,info[y],WHITE );
                y++;
        }
        vrams( 15,20," ESC : 終了 ",R_AQUA );
        vrams( 49,20," F10 : 記録開始 ",R_GREEN );
}

/*************************************************************
        日付を1日足す
*************************************************************/
void DateInc( int *yy,int *mm,int *dd )
```

```
{
        static int  monmap[]={0,31,28,31,30,31,30,31,31,30,31,30,31};
        int     d;

        if ( *mm == 2 ) {
                d = 28;
                if (*yy != 0 && (*yy % 4)==0) {
                        d++;
                }
        } else d = monmap[ *mm ];
        if ( *dd != d ) {
                *dd = *dd + 1;
                return;
        }
        *dd = 1;
        if ( *mm != 12 ) {
                *mm = *mm + 1;
                return;
        }
        *yy = *yy + 1;
        *mm = 1;
}

/*************************************************************
        時間の表示
*************************************************************/
void DispTime( receive_t *rb )
{
        uchar   s[50];
        int     jy,jm,jd,jh;
        static uchar *tmform = "%4d/%02d/%02d  %02d:%02d:%02d";

        if ( rb->hh+9 < 24 ) {
                sprintf( s,tmform,rb->yy,rb->mm,rb->dd,rb->hh+9,rb->mt,rb->ss );
        } else {
                jy = rb->yy;
                jm = rb->mm;
                jd = rb->dd;
                jh = rb->hh;
                jh = 9 - (24 - rb->hh);
                DateInc( &jy,&jm,&jd );
                sprintf( s,tmform,jy,jm,jd,jh,rb->mt,rb->ss );
        }
        vrams( 29,5,s,WHITE );
}

/*************************************************************
        衛星情報の表示
*************************************************************/
void DispEiseiInfo( receive_t *rb )
{
        uchar   s[50];
        int     i,x,y;

        for ( i=0; i<8; i++ ) {
                x = 43;
                y = i + 10;
                if( rb->eiseino[i] != INVALID ) {
                        sprintf( s,"SV-%2d    %c",rb->eiseino[i],(uchar)rb->info[i] );
                        vrams( x,y,s,WHITE );
                } else {
                        vrams( x,y,"-----     ",WHITE );
                }
        }
}

/*************************************************************
        現在位置の表示
*************************************************************/
void DispNew( receive_t *rb )
{
        uchar   s[50];
        sprintf( s,"N%3d°%02d'%02d¥"",rb->ndeg,rb->nmm,rb->nss );
        vrams( 18, 9, s,WHITE );
```

```
        sprintf( s,"E%3d°%02d'%02d¥"",rb->edeg,rb->emm,rb->ess );
        vrams( 18,10, s,WHITE );
        vrams( 19,12, rb->e,WHITE );    /* 高度 */
        vrams( 21,13, rb->f,WHITE );    /* 速度 */
        vrams( 21,14, rb->g,WHITE );    /* 方向 */
        sprintf( s,"%s 衛星測位",rb->j );
        vrams( 17,17, s,WHITE );
}

static uchar *GYOU_DAT[] = {
        "  0","+10","+20","+30","+40","+50","+60","+70","+80","+90",
        "  0","-10","-20","-30","-40","-50","-60","-70","-80","-90",
        "---",
        NULL
};
static uchar *HOU_DAT[] = {
        "   0","  +10","  +20","  +30","  +40","  +50","  +60","  +70","  +80",
        " +90"," +100"," +110"," +120"," +130"," +140"," +150"," +160"," +170",
        " -10","  -20","  -30","  -40","  -50","  -60","  -70","  -80","  -90",
        "-100","-110","-120","-130","-140","-150","-160","-170","+180",
        "----",
        NULL
};

/*************************************************************
        衛星状況の表示
*************************************************************/
void DispJyoukyou( receive_t *rb )
{
        int     i,no;

        for ( i=0; i<8; i++ ) {
                if ( rb->gyoukaku[i] >= 'A' && rb->gyoukaku[i] <= 'J' ) {
                        no = rb->gyoukaku[i] - 'A';
                } else {
                if ( rb->gyoukaku[i] >= 'a' && rb->gyoukaku[i] <= 'j' ) {
                                no = rb->gyoukaku[i] - 'a' + 10;
                        } else {
                                no = 20;
                        }
                }
                vrams( 57,i+10,GYOU_DAT[no],WHITE );    /* 仰角 */

                if ( rb->houi[i] >= 'A' && rb->houi[i] <= 'R' ) {
                        no = rb->houi[i] - 'A';
                } else {
                        if ( rb->houi[i] >= 'a' && rb->houi[i] <= 'r' ) {
                                no = rb->houi[i] - 'a' + 18;
                        } else {
                                no = 36;
                        }
                }
                vrams( 65,i+10,HOU_DAT[no],WHITE );     /* 方向 */
        }
}


/*****  GPSMON.C *****/

#include <string.h>
#include "gps.h"
#include "gpslib.h"
#include "displib.h"

/*************************************************************
        ﾓﾆﾀ表示用の窓を開く
*************************************************************/
int MonOpen( void )
{
        static uchar *msg = "123456789*123456789*123456789*123456789*123456789*123456789";

        if ( wopen( 8,1,64,23,WHITE ) == INVALID ) return INVALID;
        wputsn( 0,0,msg,WHITE,60 );
        return VALID;
```

```
}

/************************************************************
        ﾓﾆﾀ表示用の窓を閉める
************************************************************/
int MonClose( void )
{
        wclose();
        return VALID;
}

/************************************************************
        ﾓﾆﾀの表示
************************************************************/
int MonDisp( receive_t *rb )
{
        uchar s[100];

        strncpy_null( s,rb->rbuff,60 );
        wputsn( 0,1,s,AQUA,60 );
        if ( strlen(rb->rbuff) <= 60 )  return VALID;
        wputsn( 0,2,&rb->rbuff[60],AQUA,61 );
        wputsn( 0,3,"A     : ",WHITE,7 );
        wputsn( 7,3,rb->a,YELLOW,6 );
        wputsn( 0,4,"B     : ",WHITE,7 );
        wputsn( 7,4,rb->b,YELLOW,13 );
        wputsn( 0,5,"C     : ",WHITE,7 );
        wputsn( 7,5,rb->c,YELLOW,8 );
        wputsn( 0,6,"D     : ",WHITE,7 );
        wputsn( 7,6,rb->d,YELLOW,9 );
        wputsn( 0,7,"E     : ",WHITE,7 );
        wputsn( 7,7,rb->e,YELLOW,5 );
        wputsn( 0,8,"F     : ",WHITE,7 );
        wputsn( 7,8,rb->f,YELLOW,3 );
        wputsn( 0,9,"G     : ",WHITE,7 );
        wputsn( 7,9,rb->g,YELLOW,3 );
        wputsn( 0,10,"H    : ",WHITE,7 );
        wputsn( 7,10,rb->h,YELLOW,13 );
        strcpy( s,rb->i );
        strcat( s,rb->j );
        strcat( s,rb->k );
        wputsn( 0,11,"IJK  : ",WHITE,7 );
        wputsn( 7,11,s,YELLOW,3 );
        wputsn( 0,12,"L    : ",WHITE,7 );
        wputsn( 7,12,rb->l,YELLOW,5 );
        wputsn( 0,13,"M    : ",WHITE,7 );
        wputsn( 7,13,rb->m,YELLOW,5 );
        wputsn( 0,14,"N    : ",WHITE,7 );
        wputsn( 7,14,rb->n,YELLOW,5 );
        wputsn( 0,15,"O    : ",WHITE,7 );
        wputsn( 7,15,rb->o,YELLOW,5 );
        wputsn( 0,16,"P    : ",WHITE,7 );
        wputsn( 7,16,rb->p,YELLOW,5 );
        wputsn( 0,17,"Q    : ",WHITE,7 );
        wputsn( 7,17,rb->q,YELLOW,5 );
        wputsn( 0,18,"R    : ",WHITE,7 );
        wputsn( 7,18,rb->r,YELLOW,5 );
        wputsn( 0,19,"S    : ",WHITE,7 );
        wputsn( 7,19,rb->s,YELLOW,5 );
        strcpy( s,rb->t );
        strcat( s,rb->u );
        strcat( s,rb->v );
        strcat( s,rb->w );
        wputsn( 0,20,"TUVVW: ",WHITE,7 );
        wputsn( 7,20,s,YELLOW,5 );
        return VALID;
}

/*****  GPSRCV.C *****/

#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include "gps.h"
```

```
#include "gpslib.h"
#include "displib.h"
#include "sio.h"

/************************************************************
        strncpyでｺﾋﾟｰした後、最後に '¥0'を付加する
************************************************************/
char *strncpy_null( uchar *dis,uchar *sou,int len )
{
        strncpy( dis,sou,len );
        *(dis+len) = '¥0';
        return dis;
}

/************************************************************
        CR/LF まで受信する
************************************************************/
int ReceiveCR( uchar *buff )
{
        int     cnt,key,ret,tm;

        tm = tm_set( -1,300 );
        cnt = 0;
        for (;;) {
                if ( tm_get(tm) == 0 ) {
                        tm_clr(tm);
                        return -9;
                }
                key = keyin();
                if ( key == ESC ) {
                        tm_clr(tm);
                        return -1;
                }
                if ( key == F10 ) {
                        tm_clr(tm);
                        return -2;
                }
                if ( key == F6 ) {
                        tm_clr(tm);
                        return -3;
                }
                if ( key == F1 ) {
                        tm_clr(tm);
                        return -4;
                }
                ret = sio_peek();
                if ( ret == INVALID )   continue;
                *(buff+cnt) = (uchar)ret;
                cnt++;
                if ( ret == 0x0a ) break;
                if ( cnt >= MAXREC ) {
                        cnt = 0;
                }
        }
        *(buff+cnt) = '¥0';
        tm_clr(tm);
        return cnt;
}

/************************************************************
        衛星番号を返す  1～32
                        -1 : 番号なし
************************************************************/
int EiseiNo( int no )
{
        if ( no >= 'A' && no <= 'X' ) {
                return ( no - 'A' + 1 );
        }
        if ( no >= 'a' && no <= 'h' ) {
                return ( no - 'a' + 25 );
        }
        return INVALID;
}
```

```
/************************************************************
        受信ﾃﾞｰﾀの判定
************************************************************/
int RxJudge( receive_t *rb )
{
        uchar   s[80],*p[8];
        int     ret,i;

        ret = strlen( rb->rbuff );
        if ( ret <108 || ret > 110 )            return -1;
        if ( strncmp(rb->rbuff,"SONY",4) != 0 ) return -2;
        p[0] = rb->l;
        p[1] = rb->m;
        p[2] = rb->n;
        p[3] = rb->o;
        p[4] = rb->p;
        p[5] = rb->q;
        p[6] = rb->r;
        p[7] = rb->s;
        strncpy_null( rb->a,&rb->rbuff[0],6 );
        strncpy_null( rb->b,&rb->rbuff[6],13 );
        strncpy_null( rb->c,&rb->rbuff[19],8 );
        strncpy_null( rb->d,&rb->rbuff[27],9 );
        strncpy_null( rb->e,&rb->rbuff[36],5 );
        strncpy_null( rb->f,&rb->rbuff[41],3 );
        strncpy_null( rb->g,&rb->rbuff[44],3 );
        strncpy_null( rb->h,&rb->rbuff[47],13 );
        strncpy_null( rb->i,&rb->rbuff[60],1 );
        strncpy_null( rb->j,&rb->rbuff[61],1 );
        strncpy_null( rb->k,&rb->rbuff[62],1 );
        for ( i=0; i<8; i++ ) {
                strncpy( p[i],&rb->rbuff[63+(5*i)],5 );
                *(p[i]+5) = '¥0';
        }
        strncpy_null( rb->t,&rb->rbuff[103],1 );
        strncpy_null( rb->u,&rb->rbuff[104],1 );
        strncpy_null( rb->v,&rb->rbuff[105],2 );
        strncpy_null( rb->w,&rb->rbuff[107],1 );
        strncpy( s,&rb->b[0],2 );       /* 年 */
        rb->yy = atoi(s)+1900;
        if ( rb->yy > 9999 )                    return -2;
        strncpy( s,&rb->b[2],2 );       /* 月 */
        rb->mm = atoi(s);
        if ( rb->mm < 1 || rb->mm > 12 )        return -2;
        strncpy( s,&rb->b[4],2 );       /* 日 */
        rb->dd = atoi(s);
        if ( rb->dd < 1 || rb->dd > 31 )        return -2;
        strncpy( s,&rb->b[7],2 );       /* 時 */
        rb->hh = atoi(s);
        if ( rb->hh < 0 || rb->hh > 23 )        return -2;
        strncpy( s,&rb->b[9],2 );       /* 分 */
        rb->mt = atoi(s);
        if ( rb->mt < 0 || rb->mt > 59 )        return -2;
        strncpy( s,&rb->b[11],2 );      /* 秒 */
        rb->ss = atoi(s);
        if ( rb->ss < 0 || rb->ss > 59 )        return -2;

        for ( i=0; i<8; i++ ) {
                rb->eiseino[i] = EiseiNo( (int)*(p[i]+0) );
                rb->gyoukaku[i] = (int)*(p[i]+1);
                rb->houi[i] = (int)*(p[i]+2);
                rb->info[i] = (int)*(p[i]+3);
                rb->level[i] = (int)*(p[i]+4);
        }

        strncpy_null( s,&rb->c[1],2 );
        rb->ndeg = atoi( s );
        strncpy_null( s,&rb->c[3],2 );
        rb->nmm = atoi( s );
        strncpy_null( s,&rb->c[5],3 );
        if ( isalpha( (char)rb->v[0] ) ) {
                rb->nss = atoi(s) / 10;
        } else {
                rb->nss = (int)(60.0 * ( (double)atoi(s) * 0.001 ));
```

```
        }

        strncpy_null( s,&rb->d[1],3 );
        rb->edeg = atoi( s );
        strncpy_null( s,&rb->d[4],2 );
        rb->emm = atoi( s );
        strncpy_null( s,&rb->d[6],3 );
        if ( isalpha( (char)rb->v[1] )  ) {
                rb->ess = atoi(s) / 10;
        } else {
                rb->ess = (int)(60.0 * ( (double)atoi(s) * 0.001 ));
        }
        return VALID;
}


#define VALID   0
#define INVALID -1
#define ON      1
#define OFF     0
#define ANKALL  0
#define NUMONL  1
#define BEEP()  (putchar(7))

#define BLACK           0x01
#define BLUE            0x21
#define RED             0x41
#define MAGENTA         0x61
#define GREEN           0x81
#define AQUA            0xa1
#define YELLOW          0xc1
#define WHITE           0xe1

#define R_BLACK         (0x05)
#define R_BLUE          (0x25)
#define R_RED           (0x45)
#define R_MAGENTA       (0x65)
#define R_GREEN         (0x85)
#define R_AQUA          (0xa5)
#define R_YELLOW        (0xc5)
#define R_WHITE         (0xe5)

#define B_BLACK         (0x07)
#define B_BLUE          (0x27)
#define B_RED           (0x47)
#define B_MAGENTA       (0x67)
#define B_GREEN         (0x87)
#define B_AQUA          (0xa7)
#define B_YELLOW        (0xc7)
#define B_WHITE         (0xe7)

#define VF1             (0x1052)
#define VF2             (0x1053)
#define VF3             (0x1054)
#define VF4             (0x1055)
#define VF5             (0x1056)
#define F1              (0x1062)
#define F2              (0x1063)
#define F3              (0x1064)
#define F4              (0x1065)
#define F5              (0x1066)
#define F6              (0x1067)
#define F7              (0x1068)
#define F8              (0x1069)
#define F9              (0x106A)
#define F10             (0x106B)
#define F11             (0x1082)
#define F12             (0x1083)
#define F13             (0x1084)
#define F14             (0x1085)
#define F15             (0x1086)
#define F16             (0x1087)
#define F17             (0x1088)
#define F18             (0x1089)
```

```
#define F19             (0x108A)
#define F20             (0x108B)
#define ROLLUP          (0x1036)
#define ROLLDOWN        (0x1037)
#define HOME            (0x103E)
#define HELP            (0x103F)
#define INS             (0x1038)
#define DEL             (0x1039)
#define UP              (0x103A)
#define DOWN            (0x103D)
#define LEFT            (0x103B)
#define RIGHT           (0x103C)
#define NFER            (0x1057)
#define XFER            (0x1035)
#define BS              (0x08)
#define ESC             (0x1b)
#define CR              (0x0d)
#define TAB             (0x09)


 /***** GPS.H *****/

#define uchar   unsigned char
#define uint    unsigned int
#define MAXREC 110
typedef struct {
        uchar rbuff[MAXREC+50];
        uchar a[20];
        uchar b[20];
        uchar c[20];
        uchar d[20];
        uchar e[20];
        uchar f[20];
        uchar g[20];
        uchar h[20];
        uchar i[20];
        uchar j[20];
        uchar k[20];
        uchar l[20];
        uchar m[20];
        uchar n[20];
        uchar o[20];
        uchar p[20];
        uchar q[20];
        uchar r[20];
        uchar s[20];
        uchar t[20];
        uchar u[20];
        uchar v[20];
        uchar w[20];
        int     yy;
        int     mm;
        int     dd;
        int     hh;
        int     mt;
        int     ss;
        int     eiseino[8];
        int     gyoukaku[8];
        int     houi[8];
        int     info[8];
        int     level[8];
        int     ndeg;
        int     nmm;
        int     nss;
        int     edeg;
        int     emm;
        int     ess;
 } receive_t;

 extern void     DispTime( receive_t *rb );
 extern void     DispEiseiInfo( receive_t *rb );
 extern void     DispNew( receive_t *rb );
 extern void     DispJyoukyou( receive_t *rb );
 extern void     InfoDisp( void );
```

```
extern void     JyoukyouLog( receive_t *rb,char *str );
extern int      MonClose( void );
extern int      MonDisp( receive_t *rb );
extern int      MonOpen( void );
extern int      ReceiveCR( uchar *buff );
extern int      RxJudge( receive_t *rb );

extern char     *strncpy_null( uchar *dis,uchar *sou,int len );


/***** GPSLIV.H *****/

#define uchar   unsigned char
#define uint    unsigned int

extern int      clr_box( int x,int y,int w,int h,uchar c );
extern void     cursor_sw( int onoff );
extern void     cursor_position( int *x,int *y );
extern void     deb_disp( char *s,int len );
extern int      get_box( int x,int y,int w,int h,uchar *buff );
extern int      inkey( void );
extern int      inputn( int x,int y,char *kbuff,int len,int numsel );
extern int      keyin( void );
extern int      keyinp( void );
extern void     locate( int x,int y );
extern int      menusels( int x,int y,int no,uchar *str[] );
extern void     messagetm( int x,int y, char *msg,int t );
extern int      posget( int *x0,int *y0,int *x1,int *y1 );
extern int      put_box( int x,int y,int w,int h,uchar *buff );
extern int      readatr( int x,int y,int w,int h,uchar *buff );
extern void     vramfill( uchar c,int color );
extern int      vramfillxy( int x,int y,int w,int h,uchar c,int color );
extern int      vramc( int x,int y,uchar c,uint color );
extern int      vramk( int x,int y,uint sjis,uint color );
extern int      vrams( int x,int y,uchar *p,uint color );
extern int      vramsn( int x,int y,uchar *p,uint color,int len );
extern int      viskanji( int x,int y );
extern int      waku( int color );
extern int      wakuxy( int x,int y,int w,int h,int color );
extern int      wakuxyk( int x,int y,int w,int h,int color );
extern void     wclose( void );
extern int      wcls( int color );
extern void     wend( void );
extern int      winit( void );
extern int      wopen( int x,int y,int w,int h,int color );
extern int      wputs( int line,uchar *msg,int color );
extern int      wputsn( int xx,int line,uchar *msg,int color,int len );
extern int      wputsxy( int x,int y,uchar *msg,int color );

extern void     ctrl8253( int sel );
extern void     rts( int sel );
extern int      rxsio( int );
extern void     rxempty( void );
extern void     set8253( int t );
extern int      sio_emp( void );
extern void     sio_iend( void );
extern void     sio_init( int,int,int,int );
extern int      sio_peek( void );
extern void     tm_close( void );
extern int      tm_clr( int sel );
extern uint     tm_get( int sel );
extern void     tm_open( void );
extern int      tm_set( int sel,uint t );
extern void     tm_wait( uint tm );
extern int      txsio( char,int );


/***** GLIB.H *****/

extern void setb(int adr,int dat);
extern void setw(int adr,int dat);
extern void lio(int intno);
extern void ginit(void);
extern void view(double x1,double y1,double x2,double y2);
```

```
extern void window(double x1,double y1,double x2,double y2);
extern void screen(int a,int b,int c,int d);
extern void cls(int n);
extern void color(int a,int b,int c);
extern void palete(int a,int b);
extern void pset(double x,double y,int c);
extern void line(double x0,double y0,double x1,double y1,int c,int f,...);
extern void circle(double x0,double y0,double r,int c,int f,double t,...);
extern void paint(double x,double y,int c1,int c2);
extern void tile(double x,double y,int c,unsigned char *dat,int leng);
extern void get(int x1,int y1,int x2,int y2,unsigned char *dat,int leng);
extern void put(int x,int y,unsigned char *dat,int leng,
                int mode,int sw,int c1,int c2);
extern void kanji(int x,int y,unsigned int code,
                int mode,int sw,int c1,int c2);
extern void roll(int a,int b,int f);
extern int getpalete(int x,int y);
extern void move(double l,int c);
extern void setpoint(double lpx,double lpy);
extern void setangle(double a);
extern void turn(double a);
extern void moveto(double x,double y,int c);


/***** SIO.H *****/

#define B75      0
#define B150     1
#define B300     2
#define B600     3
#define B1200    4
#define B2400    5
#define B4800    6
#define B9600    7
#define B19200   8
#define B38400   9
#define BIT5     0
#define BIT6     1
#define BIT7     2
#define BIT8     3
#define PN       0
#define PO       1
#define PE       3
#define S10      1
#define S15      2
#define S20      3
#define ON       1
#define OFF      0
#define INVALID -1
#define VALID    0
#define NUL      0x00
#define SOH      0x01
#define STX      0x02
#define ETX      0x03
#define EOT      0x04
#define ENQ      0x05
#define ACK      0x06
#define DLE      0x10
#define NAK      0x15
#define SYN      0x16
#define CAN      0x18
```