

## Instituto de astronomía

### Publicaciones Técnicas



---

**“Comunicación Interna”**

**CI-2003-08**

**CONEXIÓN DEL RADIÓMETRO DE 210 GHz DEL OAN-SPM  
A TRAVÉS DE UN SERVIDOR SERIE.**

E. Colorado.

---



# **CONEXIÓN DEL RADIÓMETRO DE 210 GHz DEL OAN-SPM A TRAVÉS DE UN SERVIDOR SERIE.**

Septiembre de 2003.

*ENRIQUE COLORADO ORTIZ  
DAVID HIRIART*

Departamento de Instrumentación Electrónica  
Instituto de Astronomía, UNAM. Ensenada, B.C.  
Observatorio Astronómico Nacional, San Pedro Mártir, B.C.

## CONTENIDO

*I.- INTRODUCCIÓN.*

*II.- OBJETIVO.*

*III.- INSTALACIÓN ELECTRICA.*

*IV.- INSTALACIÓN DEL PROGRAMA DE CONTROL Y  
ADQUISICIÓN.*

*V.- CONFIGURACIÓN DEL SERVIDOR SERIE MSS1.*

*VI. VERIFICACIÓN DEL SISTEMA.*

*VII.- DIAGRAMA DE FLUJO DEL PROGRAMA DE ADQUISICIÓN.*

*VIII.- REFERENCIAS BIBLIOGRÁFICA.*

*IX.-APENDICE A.- LISTADO DEL PROGRAMA.*

## LISTA DE FIGURAS

<b>Figura No.</b>	<b>Descripción</b>
1	Diagrama de conexiones eléctricas entre el radiómetro OAN-SPM y el servidor serie Lantronix.
2	Diagrama de flujo del programa de control

## *I.- INTRODUCCIÓN.*

El radiómetro de 210 GHz del OAN-SPM es un instrumento para determinar el vapor de agua en la atmósfera a través de mediciones de la opacidad del cielo a esa frecuencia. Además de caracterizar constantemente la opacidad milimétrica de la atmósfera de la sierra de San Pedro Mártir, es utilizado como un indicador de la calidad del cielo en observaciones en el infrarrojo ya que el vapor de agua que afecta las observaciones infrarrojas no puede ser detectado visualmente. Una descripción detallada del radiómetro puede encontrarse en el Reporte Técnico RT-2001-03 del IAUNAM y en Hiriart et al. (1998). Los datos adquiridos de opacidad se encuentran en el servidor Haro con dirección electrónica

<http://haro.astrossp.unam.mx/~hiriart/radiometro>

Se han adquirido datos de San Pedro Mártir con este radiómetro desde hace aproximadamente 10 años y han sido reportados en varios artículos (Hiriart et al. 1997, Hiriart 2003). Algunos datos no han sido adquiridos principalmente por descomposturas del equipo, siendo el principal problema las descargas eléctricas en el lugar. Durante el verano del 2003 una tormenta eléctrica dañó los puertos serie de la computadora Haro donde estaba conectado el radiómetro. Para volver más versátil el instrumento sin hacer grandes modificaciones decidimos conectarlo directamente a la red internet usando un servidor serie tipo Lantronix.

## *II.- OBJETIVO.*

El objetivo de este documento es la descripción de los pasos de instalación del hardware y software de un servidor internet de puerto serie Lantronix para la comunicación y control del radiómetro de 210 GHz del OAN-SPM. Esto permitirá controlar el radiómetro desde cualquier computadora con acceso a internet. Se describen los pasos para configurar el servidor de puerto serie Lantronix. En el apéndice A se presenta un listado del programa de control que fue modificado para incluir las instrucciones del servidor de puerto serie.

## *III.- INSTALACIÓN ELECTRICA.*

La Figura 1 muestra el diagrama de conexiones eléctricas entre el radiómetro del OAN-SPM y el servidor de puerto serie Lantronix. La configuración del puerto serie del radiómetro es 8 bits, No paridad, 1 bit de alto y 2400 de baud rate.

A continuación se describen los pasos de la instalación eléctrica y de comunicación del radiómetro.

1. Conectar la alimentación de 120V CA del radiómetro mediante el cable con clavija y conector circular militar de 3 terminales.

2. Interconectar los puertos serie del radiómetro y el servidor Lantronix usando el cable con conector militar circular de 4 terminales y conector DB9.
3. Interconectar en el radiómetro los conectores circulares, macho y hembra, de tipo militar de 8 terminales con el cable apropiado. Esto alimenta la circuitería del radiómetro con las fuentes de alimentación internas en lugar de fuentes externas como pueden ser baterías o paneles solares.
4. Conectar cable de red del servidor serie Lantronix al hub de la red 132.248.\*.\*.
5. Proporcionar la alimentación del servidor Lantronix vía transformador de pared de 6VDC.

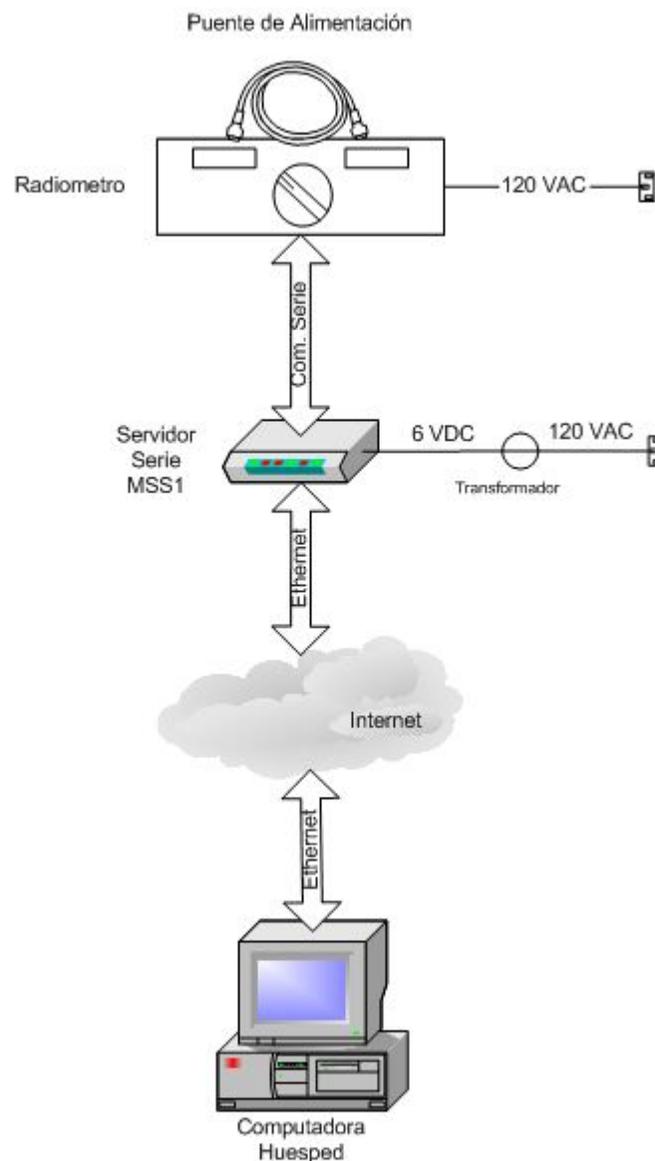


Figura 1.- Diagrama de conexiones eléctricas entre el radiómetro OAN-SPM y el servidor serie Lantronix.

#### IV.- INSTALACIÓN DEL PROGRAMA DE CONTROL Y ADQUISICIÓN.

El programa de control, adquisición y almacenamiento de datos survey reside en la computadora hoesped fue modificado de manera que pueda comunicarse con el radiómetro a través del servidor Lantronix. Este es un programa en lenguaje C cuyo listado se muestra en el Apéndice A.

A continuación se detallan los pasos a seguir cuando se instala el programa por primera vez:

1. Descomprimir el archivo fuente :  
tar xvzf survey-3.0.tar.gz
2. Dentro del directorio (cd /survey-3.0) teclear lo siguiente:  
./configure

Este comando se encarga de verificar que existan todas las bibliotecas necesarias para compilar el programa además verifica sus trayectorias.

./make

Este comando compilara y generara el archivo ejecutable.

3. El archivo ejecutable estará en /src y se llamara survey.

#### V.- CONFIGURACIÓN DEL SERVIDOR SERIE MSS1.

A continuación se describen los pasos para configurar el servidor serie MSS1-T2 marca Lantronix.

Configuración de los parámetros de la red.

- Presionar el botón de reset y estando presionado el botón, remover el voltaje de alimentación y volver a conectarlo. Al hacer esto, el dispositivo estará con todos sus parámetros establecidos por el fabricante.
- Conectar el puerto serie de una PC al puerto serie del dispositivo. Usar el Null MODEM.
- Utilice un programa de emulación de terminal, tal como minicom. Configurar el puerto a usar a 9600, 8, N, 1.
- Dar “Enter” para entrar a la consola del dispositivo y así poder configurar sus parámetros. El prompt regresado por el servidor y desplegado en la pantalla será “Local>”.
- En “username>” dar cualquier nombre.
- Para entrar como súper usuario teclear SET PRIVILEGED como password : SYSTEM.
- Cambiar la configuración a la red usada tecleando lo siguiente:

```
CHANGE IPADDRESS 132.248.4.211
CHANGE SUBNET MASK 255.255.255.0
CHANGE NAMESERVER 132.248.3.38
CHANGE DOMAIN astrossp.unam.mx
CHANGE GATEWAY 132.248.4.254
```

Configuración de los parámetros adicionales:

- Teclar las siguientes instrucciones:

```
CHANGE ACCESS REMOTE
CHANGE BOOTP DISABLED
CHANGE RARP DISABLED
```

Configuración de los parámetros del puerto serie:

- Teclar las siguientes instrucciones:

```
CHANGE SPEED 2400
```

El resto de la configuración serie es la adecuada para el radiómetro.

- Para salir de súper usuario teclear LOGOUT

Se recomienda remover la fuente de alimentación por unos segundos para volver inicializar el dispositivo con los nuevos parámetros.

## VI. VERIFICACIÓN DEL SISTEMA.

Para verificar que la configuración de la red del dispositivo esta funcionando se puede hacer lo siguiente:

- Teclar lo siguiente en el emulador de terminal para determinar si el dispositivo esta bien configurado:

```
Ping 132.248.4.211
```

- Teclar lo siguiente si el comando ping funciona correctamente:

```
telnet 132.248.4.211
```

- Y para ver la configuración de la red:

```
cualquier_nombre
```

```
show server
```

## VII.- DIAGRAMA DE FLUJO DEL PROGRAMA DE ADQUISICIÓN.

Esta versión (3.0) del programa fue modificada para que la comunicación sea vía Ethernet usando el puerto 3001. El programa actúa como cliente del servidor serie MSS1 con IP 132.248.4.211.

El programa realiza las siguientes funciones:

- Establece comunicación con el servidor MSS1 usando el protocolo TPC/IP.
- Manda un control-c para interrumpir cualquier proceso del radiómetro y que éste quede en espera de nuevas instrucciones.
- Manda el comando "RUN" para iniciar el programa del radiómetro.
- Manda la hora y el día juliano actualizados.
- Crea el archivo de datos en disco duro con formato "suYYMMDD". Donde YY es el año, MM el mes y DD el día.
- Manda el comando "A" para iniciar la adquisición.
- Captura los datos y los va almacenando en el archivo de datos.

El programa `survey` se ejecuta como una tarea de fondo en la computadora huésped. Para ejecutar el programa se deberá correr de la siguiente manera:

```
survey &
```

Cuando se ejecuta desde la consola con la opción de línea `DEBUG` entonces el programa imprime algunos mensajes que ayudan a determinar el funcionamiento del sistema:

```
survey DEBUG
```

En la figura 2 se presenta el diagrama de flujo del programa de control y adquisición `survey.c` (ver apéndice A para ver código en lenguaje C).

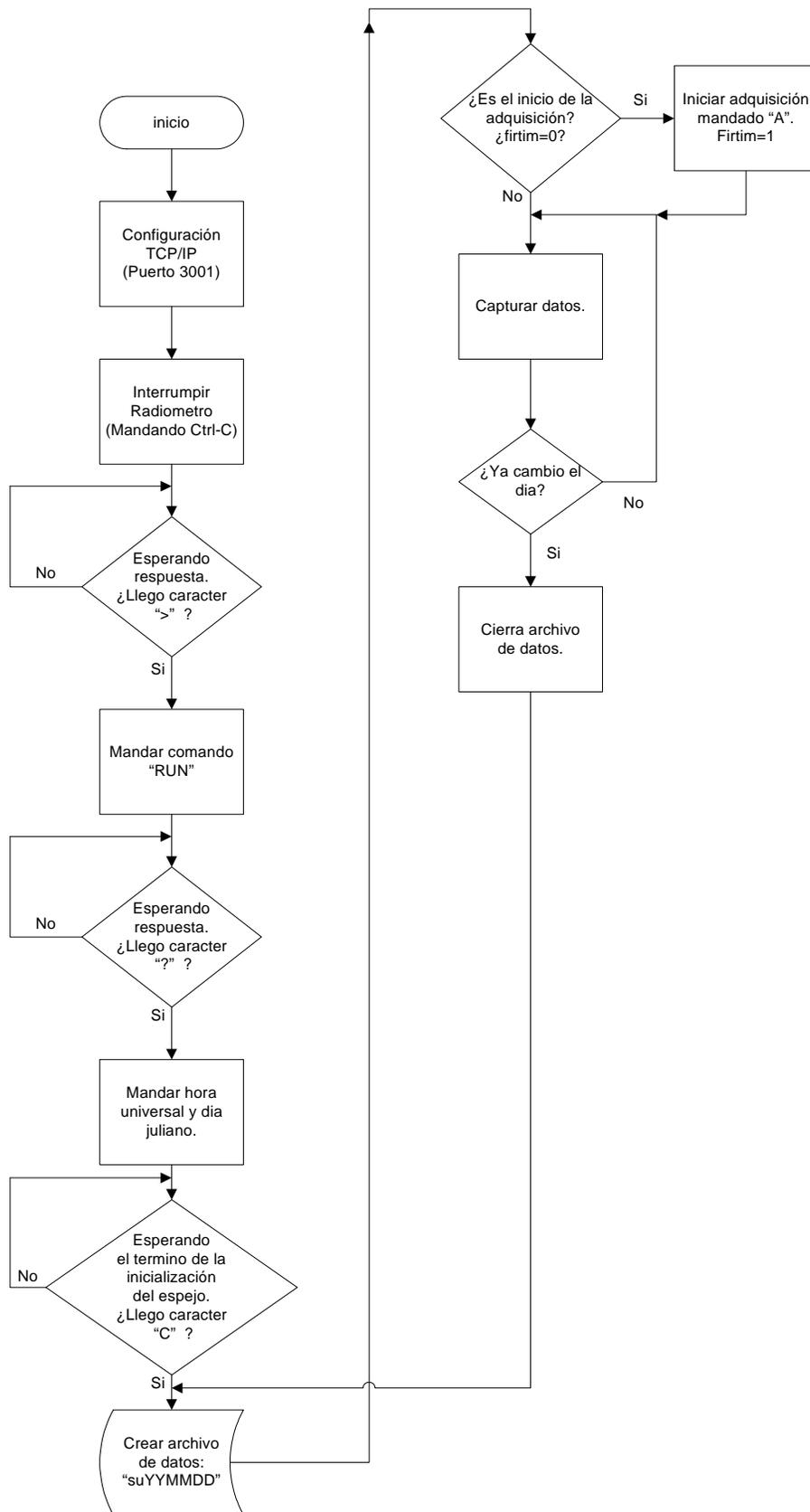


Figura 2.- Diagrama de flujo del programa de control.

### *VIII.- REFERENCIAS*

- Hiriart, D., “ El radiómetro a 210 GHz del OAN-SPM”, Reporte Técnico RT-2001-03 IAUNAM
- Hiriart, D., Goldsmith, P.F., Skrutskie, M., & Salas, L. , 1997, Rev. Mex. Astron. Astrof. Vol. 33, No.1, p. 59
- Hiriart, D., 2003, Rev. Mex. Astron. Astrof., Vol. 30, No. 1, p. 110
- ‘Begining Linux Programming’. Richard Stones y Neil Matthew. Wrox Press. Noviembre 1999.
- ‘MSS1 micro serial server installation guide’. Lantronix. Mayo 1997.

## IX.-APENDICE A.

Listado del programa survey.c:

```

/*-----
                210 GHz Site Survey Radiometer
                Five College Radio Astronomy Observatory
                University of Massachusetts at Amherst

                David Hiriart & Luis Salas
                Sept-1992
                Nov -1992
                (DOS Version)
                David Hiriart & Douglas McGonagle
                Oct-94
                (SUN Version)
                David Hiriart
                Sept-2001
                Lynux Version)
                David Hiriart & Enrique Colorado
                Halloween-2003
                (Linux TCP/IP Version V3.0)

                -----*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <time.h>
#include <termio.h>
#include <signal.h>
#include <ctype.h>
#include <sys/types.h> //librerias de tcp/ip
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#define  IGREG  (15+31L*(10+12L*1582))

void  cbrk_handler();
void  init(void);
unsigned char  readbyte(void);

struct termio  save, term;
FILE  *fp, *fps, *fpo;
int  sockfd, len, result, i, rx, PUERTO, debug=0;
char  IP[20];

main(int  argc, char  *argv[])
{
    int  day, daysent, firtim, mm, dd, yy, hh, min, sec;
    char  c, name[9], *buf = "\n", dum[79];
    long  int  refday, julday();
    time_t  now;
    struct  tm  *tmstruct;

```

```

    if (!strcmp(argv[1], "DEBUG")) {debug=1;printf("Debug mode
on...\n");}
    PUERTO=3001;
    strcpy(IP, "132.248.4.211");
    init();      //inicializacion de la com. via red.

    /* Initiate CNTRL-C handler */
    if(signal(SIGINT, cbrk_handler) == SIG_ERR) {
        perror("\n ERROR: Signal Failed.\n\n");
        exit(0);
    }
    if (debug ) {
        printf("\n LMT 210 GHz SITE SURVEY RADIOMETER DATA LOGGER
REV. 3.0\n");
        printf(" University of Massachusetts, FCRAO @ 1994\n");
        printf("Asegurarse de que el radiometro ya este encendido!\n");
    }
    firtim = 0;
    daysent =0;
    c=toascii(3);
    if (debug) printf("Voy a mandar control-break\n");
    fprintf(fpo, "%c\r", c);
    fflush(fpo);
    if (debug) printf("Ya mande control-break\n");
    sleep(1);
    for(;;){

    /* CNTRL-C the radiometer and gets its attention */
    fprintf(fpo, "%c", toascii(3));
    fflush(fpo);
    do{
        if (debug) printf("Waiting.....\n");
        read(fileno(fps), &c, sizeof(char));
        if (debug) printf("Esperando ">" ,RX=%c\n",c);

    }while(c != '>');

    /* Send RUN command and start the radiometer firmware */
    fprintf(fpo, "RUN\r\n");
    fflush(fpo);
    if (debug) printf("mandando RUN\n",c);

    /* Wait for firmware prompt (?) */
    do{
        read(fileno(fps), &c, sizeof(char));
        if (debug) printf("Esperando "?" ,RX=%c\n",c);
    }while(c != '?');

    /* Get time and Julian date, send it to the radiometer */
    time(&now);
    tmstruct = localtime(&now);
    hh = tmstruct->tm_hour;
    min = tmstruct->tm_min;
    sec = tmstruct->tm_sec;
    mm = tmstruct->tm_mon+1;
    dd = tmstruct->tm_mday;
    yy = tmstruct->tm_year+1900;

```

```

    refday=julday(mm,dd,yy)-julday(1,1,1992)+1L;
    fprintf(fpo, "%d,%d,%d,%d\r\n", hh, min, sec, refday);
    fflush(fpo);

/*Wait for the radiometer to home the mirror, wait for prompt (C)
*/
do {
    read(fileno(fps), &c,sizeof(char));
    if (debug) printf("Esperando " "C" ",RX=%c\n",c);
} while (c != 'C');
if (debug) printf("\nDone...\n\n");

/* Create date stamp and open data file */
time(&now);
tmstruct = localtime(&now);
mm = tmstruct->tm_mon+1;
dd = tmstruct->tm_mday;
yy = tmstruct->tm_year;
    yy -= 100;
sprintf(name,"su%.2d%.2d%.2d", yy,mm,dd);
if (debug) printf(" Data File: %s\n\n",name);
if((fp = fopen(name,"a")) == NULL){
    fclose(fps);
    exit(0);
}
/* If this is the first time through loop send */
/* the radiometer A command to start data taking */
do{
    read(fileno(fps), &c, sizeof(char));
    if (debug) printf("Esperando " "?" ",RX=%c\n",c);
}while(c != '?');

if(firtim == 0){
    firtim=1;
    if (debug) printf("Mandando A\n");
    fprintf(fpo,"A\r\n");
    fflush(fpo);
}

/* Write data to file until the day changes */
do {
    read(fileno(fps), &c, sizeof(char));
    if (debug) putchar(c);
    if(c != toascii(13)) write(fileno(fp), &c, sizeof(char));

        if (c == '*') fflush(fp);
        time(&now);
        tmstruct = localtime(&now);
        day = tmstruct->tm_mday;
    }while(dd==day);
fflush(fp);
fclose(fp);
    firtim=0;
}
return 0;
}

```

```

/*****
/***** Function: Calculate Julian Day *****/
/*****
long julday(mm,id,iyyy)
int mm,id,iyyy;
{
    long jul;
    int ja,jy,jm;

    if(iyyy==0) printf("Error in JULDAY\n");
    if(iyyy<0) ++iyyy;
    if(mm>2){
        jy=iyyy;
        jm=mm+1;
    }
    else {
        jy=iyyy-1;
        jm=mm+13;
    }
    jul=(long)(floor(365.25*jy)+floor(30.6001*jm)+id+1720995);
    if(id+31L*(mm+12L*iyyy) >= IGREG ){
        ja=0.01*jy;
        jul+=2-ja+(int)(0.25*ja);
    }
    return(jul);
}
/*****
/***** Function: Break Handler (CNTRL-C) *****/
/*****
void cbrk_handler()
{
    if (fp != NULL) {
        fflush(fp);
        fclose(fp);
    }
    /* CNTRL-C the radiometer and get its attention */
    fprintf(fpo, "%c", toascii(3));
    fflush(fpo);
    if (fps != NULL) fclose(fps);
    if (fpo != NULL) fclose(fpo);
    ioctl(fileno(fps), TCSETA, &save);
    exit(0);
    signal(SIGINT, cbrk_handler);
}
//*****
void init(void)
{
    struct sockaddr_in address;
    if (debug) printf("\nhacia:%s, Puerto:%d\n",IP,PUERTO);
    sockfd=socket(AF_INET, SOCK_STREAM,0);
    address.sin_family=AF_INET;
    address.sin_addr.s_addr = inet_addr(IP);
    address.sin_port = htons(PUERTO);
    len = sizeof(address);
    result = connect (sockfd, (struct sockaddr *)&address,len);
    if (result!=-1)
    {

```

```
        perror("oops: No esta listo el puerto\n");
        printf("( %s)[%d]\n",IP,PUERTO);
    }
    fps = fdopen(sockfd,"r");
    if( (fps == NULL) ){
        close(sockfd );
        printf("-2");
        printf(" Err en el fdopen file Entrada fps\n");
        exit(0);
    }
    fpo= fdopen( dup(sockfd), "w");
    if( (fpo == NULL) ){
        printf(" Err en el fdopen file Salida fpo\n");
        printf("-2");
        exit(0);
    }
    if (debug) printf("\nFIN init cliente\n");
}
//*****
```



**Comité Editorial de Publicaciones Técnicas  
Instituto de Astronomía  
UNAM**

**M.C. Urania Ceseña  
Dr. Carlos Chavarria  
M.C. Francisco Murillo**

**Observatorio Astronómico Nacional  
Km. 103 Carretera Tijuana-Ensenada  
22860 Ensenada B.C.  
[editorial@astroesen.unam.mx](mailto:editorial@astroesen.unam.mx)**