

**Cliente en Java para
el sistema óptico de apuntado
del radiotelescopio de 40m**

L. Barbas, P. de Vicente

Informe Técnico IT-OAN 2006-9

ÍNDICE

1. EL SISTEMA ÓPTICO DE APUNTADO DEL RADIOTELESCOPIO DE 40M.....	1
1.1 <i>Introducción</i>	1
1.2 <i>Sistema óptico completo</i>	1
1.3 <i>Características de los equipos</i>	4
1.3.1 Cámara CCD Mintron.....	4
1.3.2 Enfocador TCF-S.....	5
1.4 <i>Componentes ACS</i>	5
2. CLIENTE JAVA.....	6
2.1 <i>Diseño detallado</i>	6
2.1.1 Diagrama de clases general.....	7
2.1.2 Paquete alma.opticaltel.....	8
2.1.2.1 La clase opticaltelClient.....	8
2.1.2.2 La clase mintronJClient.....	9
2.1.2.3 La clase focuserJClient.....	9
2.1.2.4 La clase CBdoubleTextFieldTemp.....	11
2.1.2.5 La clase CBlongTextFieldPos.....	12
2.1.2.6 La clase opticaltelGUI.....	13
2.1.2.7 La clase FrameRearPanel.....	14
2.2 <i>Detalles de Implementación</i>	15
2.2.1 Monitores.....	15
2.3 <i>Interfaces gráficas</i>	16
2.4 <i>Pruebas del sistema</i>	20
2.4.1 Cámara CCD Mintron.....	20
2.4.2 Enfocador TCF-S.....	22
2.5 <i>Java Web Start</i>	22
3. REFERENCIAS.....	24

1. El sistema óptico de apuntado del radiotelescopio de 40m

1.1 Introducción

Una vez finalizada la construcción del radiotelescopio de 40 m de diámetro del CAY, es necesario realizar algunas comprobaciones iniciales como son la inclinación del eje de acimut, el apuntado, el seguimiento, etc. Para verificar la puntería se instalará transitoriamente un sistema óptico de apuntado en la parte posterior del subreflector del radiotelescopio.

En algunos interferómetros milimétricos y submilimétricos se utiliza un sistema óptico de apuntado permanente, ya que es utilizado para la determinación del modelo de apuntado inicial tras cada traslado de las antenas y para la comprobación ocasional del apuntado. En algún telescopio submilimétrico se podría emplear también como sistema de autoguiado. En nuestro caso, se trata de poder determinar un apuntado preliminar de la antena cuando todavía no se tiene ninguna referencia de la orientación precisa de la antena y sin necesidad de disponer de un sistema receptor sensible instalado.

El sistema óptico constará de un telescopio óptico solidario con la estructura móvil del radiotelescopio. El telescopio óptico incorporará una cámara CCD capaz de obtener la posición de un número suficiente de estrellas, cuyas posiciones se conocen con precisión. Dado que lo que se pretende es determinar el apuntado observando un número suficiente de estrellas que se encuentren sobre el horizonte, el telescopio puede ser de pequeño tamaño. También hay que tener en cuenta que existen numerosas limitaciones, como por ejemplo que no es posible realizar el apuntado de día, salvo que la CCD sea sensible a la radiación infrarroja (NIR), que es necesario poder enfocar, especialmente si la CCD responde a un amplio rango de longitudes de onda y que no es posible realizar el apuntado si hay nubes.

1.2 Sistema óptico completo

El sistema óptico de apuntado elegido para el radiotelescopio de 40m consta de un telescopio óptico, una cámara CCD y un enfocador automatizado. El telescopio óptico es un refractor de focal corta y sin trípode modelo SkyWatcher SW1025 (Diámetro = 102 mm, focal = 500 mm, longitud > 50 cm). El telescopio es resistente a la intemperie, ya que la instalación deberá permanecer varias semanas en el subreflector.



Fig. 1. Telescopio Sky Watcher 102-500

La cámara CCD es de la marca Mintron modelo 12V1C-EX. En la parte posterior dispone de cinco pulsadores que permiten seleccionar diferentes modos de operación a través de un menú sobrepuesto en la pantalla. Este menú se activa al pulsar prolongadamente el botón central. Esta cámara genera una señal de video compuesto a través de un conector BNC y una señal de S-Video a través de un conector mini-DIN de 4 pines.

La cámara estará situada en el subreflector, un lugar de acceso complicado, por lo que es necesario poder controlarla remotamente. Para ello se han realizado algunas modificaciones en la cámara [2], se ha añadido un circuito con un microprocesador que permite simular la pulsación de los botones y que permite su operación remota. El circuito, de pequeñas dimensiones, se ha instalado en el interior de la cámara y se le ha cargado un programa en el microprocesador.



Fig. 2. Vista trasera de la cámara Mintron

El sistema de focalización es de la marca Optec modelo TCF-S (Temperature Compensating Focuser). Es un enfocador que permite enfoque manual y automático, es decir, permite mantener automáticamente el foco del telescopio compensando las dilataciones y contracciones debidas a la temperatura. Además este dispositivo es controlable a través de un puerto serie mediante el envío de comandos simples, por lo que se puede enfocar remotamente sin necesidad de estar junto al telescopio.



Fig. 3. Sistema de enfoque TCF-S

La cámara y el enfocador poseen una interfaz serie RS232 para operar remotamente sobre ellos, puesto que estos dispositivos están situados en el subreflector y los cables RS232 no pueden ser excesivamente largos, se ha utilizado un conversor Lantronix serie-ethernet modelo MSS4 con 4 puertos serie para que estos equipos estén disponibles a través de la red local.

La comunicación entre un ordenador y el sistema de enfoque TCF-S se debe realizar a 19200 baudios, con 8 bits de datos, 1 bit de parada, sin paridad, con acceso dinámico y flujo on/off por lo que el puerto del Lantronix al que se conecte el enfocador deberá estar configurado con estos valores. En el caso de la cámara Mintron únicamente varía la velocidad que será de 9600 baudios.



Fig. 4. Conversor serie-ethernet Lantronix

1.3 Características de los equipos

1.3.1 Cámara CCD Mintron

La cámara Mintron permite seleccionar diferentes opciones de configuración a través de su menú. El desplazamiento por las distintas opciones del menú se realiza con los 4 pulsadores de dirección del panel trasero o remotamente mediante el envío de los comandos que simulan la pulsación de estos botones.



Fig. 5. Menú de la cámara Mintron

La cámara permite, entre otras, las siguientes opciones:

- Establecer un título para la imagen.
- Activar/desactivar el título.
- Colocar el título en distintas posiciones.
- Seleccionar el tiempo de integración entre 1 y 128 segundos.
- Seleccionar el modo de control de luz electrónico o automático.
- Ajustar el nivel de brillo en el intervalo [1,9].
- Seleccionar el modo de ganancia: automático, manual o desactivado.
- Seleccionar el nivel de ganancia en dB.
- Mostrar la imagen en positivo o negativo.
- Crear una imagen especular de la imagen original.
- Controlar el modo de funcionamiento de la cámara en condiciones de poca visibilidad.
- Aumentar la imagen con zoom digital hasta un máximo del doble de su tamaño.
- Activar/ desactivar el zoom.
- Volver a una configuración con los valores predefinidos en fábrica.

1.3.2 Enfocador TCF-S

El enfocador consta de una caja de control que permite el control manual, automático o remoto del movimiento del motor. Esta caja está compuesta por:

- Un visor que muestra la posición del enfocador en cada momento.
- Un selector de modo de operación que permite seleccionar entre manual, automático (ajuste de posición en función de la temperatura) y remoto. Otro modo de operación posible del enfocador es bajo consumo.
- Un selector de programa que debe estar en la posición RUN para que el enfocador se pueda controlar remotamente.
- Dos botones IN y OUT que permiten mover el motor en ambas direcciones cuando el enfocador está en modo manual.

1.4 Componentes ACS

La funcionalidad de la cámara y el enfocador se ha implementado mediante clases C++. Los métodos de estas clases envían a los dispositivos aquellos comandos que han sido definidos para su control remoto.

Enfocador TCF-S	
Comando	Descripción
FMODE	Modo remoto
FAMODE	Modo automático A
FBMODE	Modo automático B
FIxxxx	Mover xxxx pasos hacia dentro
FOxxxx	Mover xxxx pasos hacia fuera
FPOSRO	Leer posición
FTMPRO	Leer temperatura
FCENTR	Centrar en 3500
FSLEEP	Modo durmiente
FWAKUP	Despertar del modo durmiente
FREADA	Leer pendiente A
FREADB	Leer pendiente B
FLAxxx	Comandar pendiente A
FLBxxx	Comandar pendiente B
FLBxxx	Modo manual
FFMODE	

Cámara MINTRON	
Comando	Descripción
U ó 8	Hacia arriba
D ó 2	Hacia abajo
L ó 4	Izquierda
R ó 6	Derecha
E ó 5	Intro
Q	Mostrar menú

Tanto la cámara como el enfocador se van a controlar a través del sistema de control remoto del radiotelescopio. Este sistema utiliza el ACS (Alma Common Software). Se ha desarrollado un componente ACS para controlar la cámara Mintron [2] y otro para controlar el enfocador TCF-S [3]. El componente de la cámara define propiedades de lectura y escritura para cada una de las opciones de configuración (título, posición título, nivel de brillo, etc), un método para simular la pulsación de los botones del panel trasero y un método para reiniciar la cámara y fijar los valores predefinidos. El componente del enfocador define métodos para establecer cada uno de los modos de operación y para mover el foco en una y otra dirección. Además define propiedades de lectura y escritura para la posición del foco y las pendientes posición/°C de los modos automáticos, y una propiedad de sólo lectura para la temperatura.

Los métodos y las propiedades de ambos componentes ACS utilizan la clase C++ para comunicarse con sus respectivos dispositivos.

Desde un cliente serán accesibles directamente cada uno de los métodos y propiedades del componente. En las propiedades en las que esté permitida la escritura se podrá introducir un valor por medio la función `set_sync()` de la propiedad. Para obtener el valor de las propiedades de lectura se utilizará la función `get_sync()` de la propiedad.

2. Cliente Java

El trabajo realizado durante el mes de prácticas ha sido diseñar y desarrollar un cliente Java para el control del sistema óptico de apuntado del radiotelescopio de 40m. Este cliente permite controlar tanto la cámara CCD como el sistema de enfoque desde cualquier ordenador del sistema de control del radiotelescopio.

2.1 Diseñodetallado

El objetivo del cliente Java es permitir la modificación de las opciones de configuración de la cámara y el enfocador, sin tener acceso manual a tales dispositivos y

de un modo sencillo. Para ello el cliente implementa una interfaz gráfica que permite interactuar con el sistema óptico de apuntado.

2.1.1 Diagrama de clases general

Los clientes que interactúan con los componentes ACS de la cámara y el enfocador se han implementado en dos clases java denominadas `mintronJClient` y `focuserJClient`, respectivamente. Estas clases heredan de `alma.acs.component.client.ComponentClient`. La comunicación entre estos clientes y los componentes se realiza a través de sus propiedades y métodos.

La interfaz gráfica principal que muestra las opciones de configuración de los dispositivos está implementada en la clase `opticaltelGUI`. Además se ha definido otra interfaz gráfica `FrameRearPanel` que simula los 5 pulsadores del panel trasero de la cámara. Ambas clases heredan de `javax.swing.JFrame`.

El punto de arranque de la aplicación es la clase `opticaltelClient` que contiene el método `main()`. Esta clase crea los clientes ACS para la cámara y el enfocador y permite la visualización de la interfaz gráfica principal.

El sistema de enfoque cuando trabaja en modo automático proporciona continuamente el valor de posición y temperatura. Ha sido necesario crear un `callback` para cada una de estas variables que permite que se refresquen los valores de temperatura y posición en la interfaz gráfica. Estas `callbacks` se implementan en la clase `CbdoubleTextFieldTemp` para la temperatura y en la clase `CblongTextFieldPos` para la posición.

Todas las clases forman parte del paquete `alma.opticaltel`. Por ello los archivos fuente `.java` deben ir situados en un directorio denominado `opticaltel` contenido en otro directorio denominado `alma`. Para compilar estos ficheros y que se generen los archivos `.class`, que se corresponden con las clases java desarrolladas, se añaden en el archivo `Makefile`, las siguientes líneas:

```
#
# Jarfiles and their directories
#
JARFILES=OpticalTelClient
OpticalTelClient_DIRS=alma
jjj_EXTRAS=
```

Será necesario colocar el directorio `alma` en el mismo lugar que se encuentre el archivo `Makefile`. Estas líneas, además, generan el fichero `OpticalTelClient.jar` que empaqueta las clases que forman el cliente.

La figura siguiente muestra el diagrama de clases general del cliente, en él aparecen las clases diseñadas y las relaciones entre ellas. Las flechas indican una relación de dependencia, es decir, una clase utiliza a otra que necesita para su realizar su cometido.

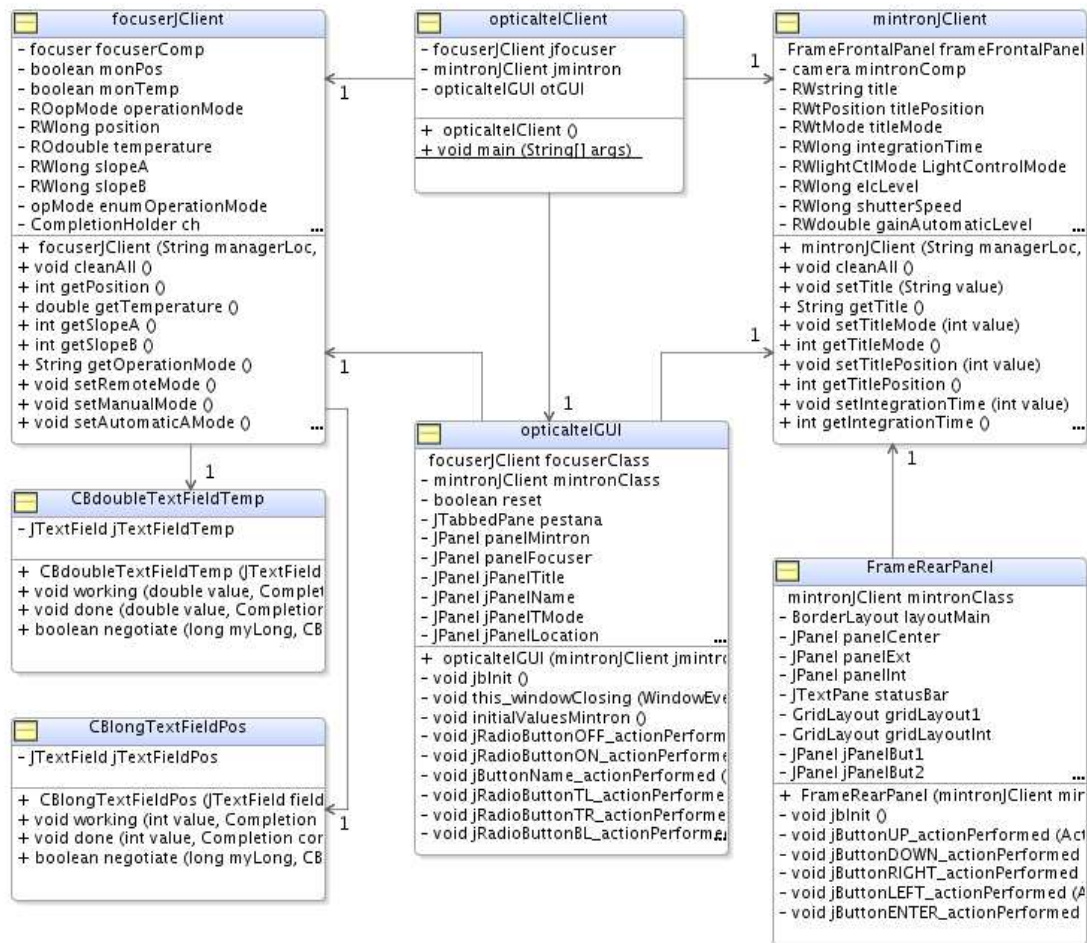


Fig. 6. Diagrama general de clases

2.1.2 Paquete alma.opticaltel

En esta sección se describen en detalle cada una de las clases del paquete `alma.opticaltel`.

2.1.2.1 La clase `opticaltelClient`

Esta clase contiene el método `main()` que permite iniciar la aplicación. Para iniciar el cliente en un ordenador que tenga instalado el software ACS, una vez que el manager y los componentes están activados, hay que ejecutar el siguiente comando:

```
acsStartJava alma.opticaltel.opticaltelClient
```

Donde el argumento pasado es el nombre de la clase que contiene el método `main()` precedido del paquete al que pertenece.

En el constructor de esta clase, llamado desde `main()`, se crea un objeto de la clase `mintronJClient` y otro de la clase `focuserJClient`. También se crea un objeto de la interfaz gráfica y se permite su visualización.

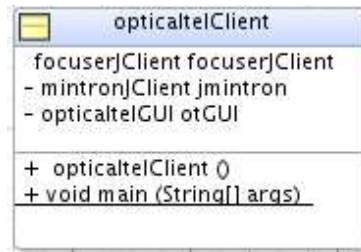


Fig. 7. La clase `opticaltelClient`

2.1.2.2 La clase `mintronJClient`

El constructor de esta clase crea un objeto del componente “MINTRON” y otro por cada una de las propiedades del componente.

Todas las propiedades son de lectura y escritura, por lo que se han definido métodos públicos `getXXX` y `setXXX` para poder obtener y establecer los valores de las propiedades desde cualquier otra clase. Del mismo modo se han implementado una funciones públicas que permiten el acceso a los métodos definidos en el componente ACS de la cámara.

El diagrama de la página siguiente muestra las variables y funciones definidas en la clase `mintronJClient`.

2.1.2.3 La clase `focuserJClient`

El constructor de esta clase crea un objeto del componente “TCFS” y otro por cada una de las propiedades del componente.

Para cada propiedad de lectura se ha definido una función pública `getXXX` que permite obtener su valor desde cualquier otra clase. Para las propiedades de escritura se han definido funciones públicas `setXXX` para establecer su valor.

Del mismo modo se han implementado funciones públicas accesibles desde cualquier clase que permiten la llamada a los métodos definidos en el componente ACS del enfocador.

En esta clase se definen dos monitores `Monitordouble monitorTemperature` y `Monitorlong monitorPosition` que están consultando constantemente (a intervalos de tiempo especificados) el valor de las propiedades de temperatura y posición del enfocador. Estos monitores se crean, activan y desactivan con las funciones

setMonitorXXX, startMonitorXXX y stopMonitorXXX, definidas en la clase. Sólo será necesario crear y activar los monitores cuando el enfocador trabaje en modo automático.

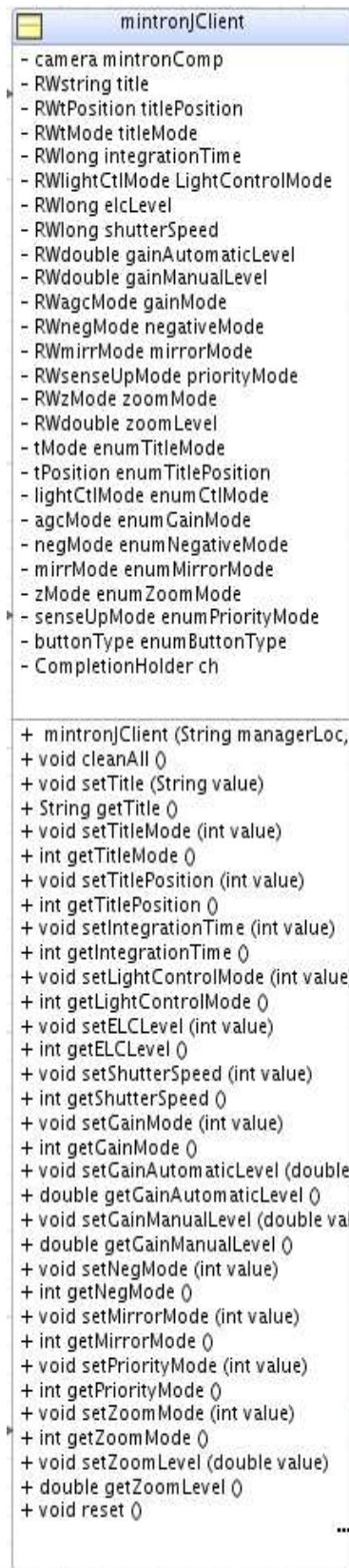


Fig. 8. La clase mintronJClient

Si el usuario selecciona o sale de modo de operación automático en la interfaz gráfica, desde la clase `opticaltelGUI` se realizará la llamada a las funciones `setMonitorXXX`, `startMonitorXXX` o `stopMonitorXXX`, según convenga.

El diagrama siguiente muestra las variables y funciones definidas en la clase `focuserJClient`.

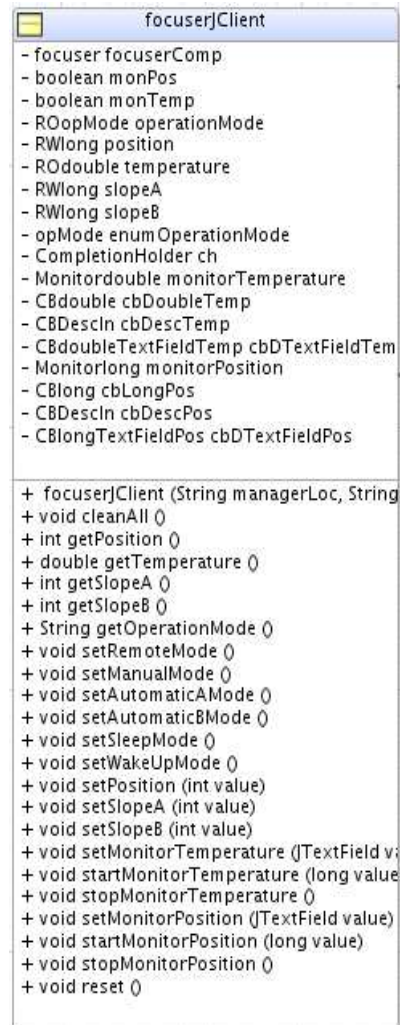


Fig. 9. La clase `focuserJClient`

2.1.2.4 La clase `CBdoubleTextFieldTemp`

La clase `CBdoubleTextFieldTemp` hereda de `alma.ACS.CBdoublePOA`. El objetivo de esta clase es actualizar en la interfaz gráfica el valor de la temperatura (tipo `double`) cuando el enfocador está en modo automático.

Esta clase se utiliza en `focuserJClient` para crear el monitor de temperatura.

La temperatura se muestra en la interfaz gráfica a partir una caja de texto (`JTextField jTextFieldTemp`), por lo que esta variable `jTextFieldTemp` se pasa como parámetro al constructor de la clase.

Los métodos `working` y `done` de la clase `CbdoublePOA` han sido reimplementados aquí. El método `working` es llamado cada vez que la propiedad asociada al monitor de temperatura modifica su valor y lo que hace es actualizar el valor de la caja de texto de temperatura. El método `done` es llamado cuando la monitorización finaliza y también actualiza el valor de la caja de texto de temperatura.

El diagrama siguiente muestra las variables y funciones definidas en la clase `CbdoubleTextFieldTemp`.

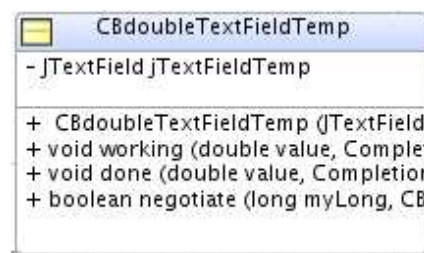


Fig. 10. La clase `CbdoubleTextFieldTemp`

2.1.2.5 La clase `CBlongTextFieldPos`

La clase `CBlongTextFieldPos` hereda de `alma.ACS.CBlongPOA`. El objetivo de esta clase es actualizar en la interfaz gráfica el valor de la posición (tipo `long`) cuando el enfocador está en modo automático.

Esta clase se utiliza en `focuserJClient` para crear el monitor de posición.

La posición se muestra en la interfaz gráfica a partir una caja de texto (`JTextField jTextFieldPos`), esta variable `jTextFieldPos` se pasa como parámetro al constructor de la clase.

Los métodos `working` y `done` de la clase `CBlongPOA` han sido reimplementados aquí. El método `working` es llamado cada vez que la propiedad asociada al monitor de posición modifica su valor y lo que hace es actualizar el valor de la caja de texto de posición. El método `done` es llamado cuando la monitorización finaliza y también actualiza el valor de la caja de texto de posición.

El diagrama siguiente muestra las variables y funciones definidas en la clase `CBlongTextFieldPos`.

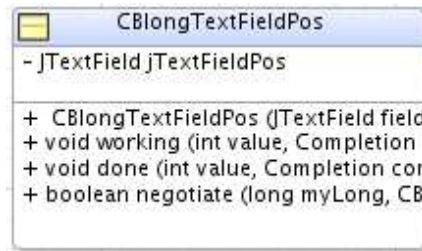


Fig. 11. La clase CblongTextFieldPos

2.1.2.6 La clase opticaltelGUI

La clase `opticaltelGUI` implementa la interfaz gráfica principal que permite al usuario realizar el control remoto sobre la cámara Mintron y el enfocador TCFS. Las opciones de configuración de cada dispositivo aparecen en una pestaña diferente. En la sección 2.3 se puede ver el formato completo de la interfaz.

El diagrama siguiente muestra las variables y funciones definidas en la clase `opticaltelGUI`.

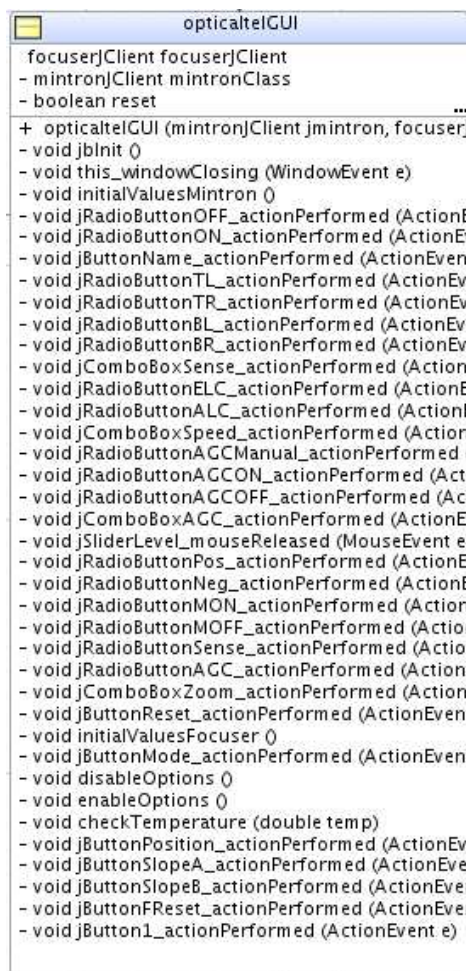


Fig. 12. La clase opticaltelGUI

Esta clase hereda de `javax.swing.JFrame`, por lo que permite presentar un *frame*, una ventana de alto nivel con título y borde.

El constructor de la clase recibe como parámetro un objeto de la clase `mintronJClient` y otro de la clase `focuserJClient` de este modo se tiene acceso completo a todos sus métodos. También se realiza una llamada al método `jbInit()`.

El método `jbInit()` contiene el código que desarrolla el formato de la ventana gráfica: pestañas, etiquetas, cajas de texto, listas desplegables, botones, etc. Los elementos asociados al valor de una propiedad implementan la interface `java.awt.event.ActionListener` de modo que estarán a la “escucha” de cualquier modificación que el usuario realice sobre ellos. Cuando esto ocurra se llamará a los métodos `XXX_actionPerformed()` encargados de modificar el valor de la propiedad correspondiente, a través de los objetos `mintronJClient` y `focuserJClient`.

2.1.2.7 La clase *FrameRearPanel*

Esta clase hereda de `javax.swing.JFrame`. Permite presentar una ventana con 5 botones (up, down, right, left, enter) que simula el panel trasero de la cámara Mintron.

El constructor de la clase recibe como parámetro un objeto de la clase `mintronJClient` de este modo se tiene acceso completo a todos sus métodos.

El método `jbInit()` contiene el código que desarrolla el formato de la ventana gráfica. Cada botón implementan la interface `java.awt.event.ActionListener` de modo que esta a la “escucha” de cualquier pulsación que el usuario realice sobre él. Cuando se pulsa uno de los botones se realiza una llamada al método `jButtonUP_actionPerformed`, `jButtonDOWN_actionPerformed`, `jButtonRIGHT_actionPerformed`, `jButtonLEFT_actionPerformed` o `jButtonENTER_actionPerformed`, según corresponda. Estos métodos enviarán el comando correspondiente a la cámara por medio del objeto `mintronJClient`.

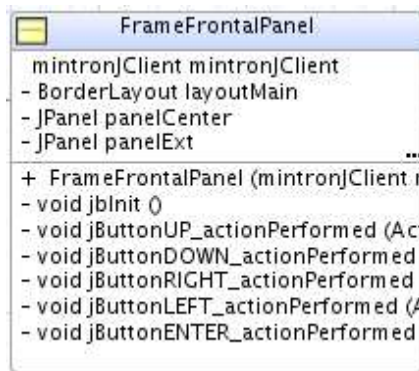


Fig. 13. La clase *FrameRearPanel*

2.2 Detalles de Implementación

2.2.1 Monitores

En esta sección se muestra el código detallado de la creación de los Monitores. Nos centraremos en el caso del Monitor de posición. (El Monitor de temperatura es totalmente análogo).

El sistema de enfoque en los modos automáticos A y B proporciona continuamente el valor de la posición del foco y la temperatura. En esta situación las propiedades de posición y temperatura del componente ACS están continuamente actualizándose. Para que el cliente pueda obtener los valores de estas variables es necesario crear un Monitor.

Los Monitores reciben como parámetro un callback que se ejecuta cada vez que la propiedad que está asociada al Monitor cambia de valor. Este callback será el encargado de modificar la interfaz gráfica para que el usuario visualice los valores de posición y temperatura actualizados en todo momento.

Cuando en la interfaz gráfica (clase `opticaltelGUI`) el usuario selecciona un modo automático para el enfocador se realiza una llamada a los métodos `setMonitorPosition()` y `startMonitorPosition()` de la clase `focuserJClient`.

```
// Clase focuserJClient

private Monitorlong monitorPosition;
private CBlong cbLongPos;
private CBDescIn cbDescPos = new CBDescIn();
private CBlongTextFieldPos cbDTextFieldPos;

...

public void setMonitorPosition(JTextField value){
    try{
        cbDTextFieldPos = new CBlongTextFieldPos(value);
        cbLongPos=CBlongHelper.narrow
            (getContainerServices().activateOffShoot(cbDTextFieldPos));
        monitorPosition = position.create_monitor(cbLongPos,cbDescPos);
        monPos = true;

    }catch(Exception excep) {
        excep.printStackTrace();
    }
}

public void startMonitorPosition(long value){
    monitorPosition.set_timer_trigger(value); // 10000000 = 1 s (Time = 10 ns)
}

public void stopMonitorPosition(){
    monitorPosition.set_timer_trigger(0); // Timer off
}
```

El método `setMonitorPosition()` crea el Monitor asociado a la propiedad de posición (`position.create_monitor()`) pasando como parámetro el callback `cbLongPos` que será llamado cada vez que la propiedad modifique su valor. Además

para que la interfaz gráfica esté siempre actualizada el callback `cbLongPos` se asocia a la clase `CBLongTextFieldPos` creada para este cometido (ver sección 2.1.2.5).

```
// Clase CBLongTextFieldPos
public class CBLongTextFieldPos extends CBLongPOA{

    private JTextField jTextFieldPos;
    public CBLongTextFieldPos(JTextField field) {
        jTextFieldPos = field;
    }

    public void working(int value, Completion completion, CBDescOut desc){
        jTextFieldPos.setText((new Integer(value)).toString());
    }

    public void done(int value, Completion completion, CBDescOut desc){
        jTextFieldPos.setText((new Integer(value)).toString());
    }

    public boolean negotiate(long myLong, CBDescOut desc){
        return true;
    }
}
```

El método `startMonitorPosition()` establece el intervalo de disparo del monitor en centenas de nanosegundos.

El método `stopMonitorPosition()` establece el intervalo de disparo del monitor a cero, por lo que el monitor queda desactivado (no destruido). Este método será llamado cuando el usuario saque el enfocador de modo automático.

2.3 Interfaces gráficas

Las figuras siguientes muestran el cliente gráfico del sistema óptico de apuntado del radiotelescopio de 40m. La interfaz “40 m Optical Telescope” dispone de dos pestañas, una para la cámara Mintron y otra para el enfocador TCF-S. Al iniciar la aplicación se muestran en pantalla los valores que tiene configurado el sistema en ese momento.

El formato de la pestaña de la cámara Mintron se muestra en la figura de la página siguiente. Todas las opciones de configuración de la cámara se pueden modificar a través de esta interfaz:

Título

- ✓ Establecer un título para la imagen -> Escribir título y pulsar botón *Set*.
- ✓ Activar/desactivar el título -> *Mode ON/OFF*
- ✓ Colocar el título en distintas posiciones->En *Location* (Sólo activado cuando hay un título establecido) seleccionar:
 - *TR* (top right)
 - *TL* (top left)
 - *BR* (botton right)
 - *BL* (botton left)

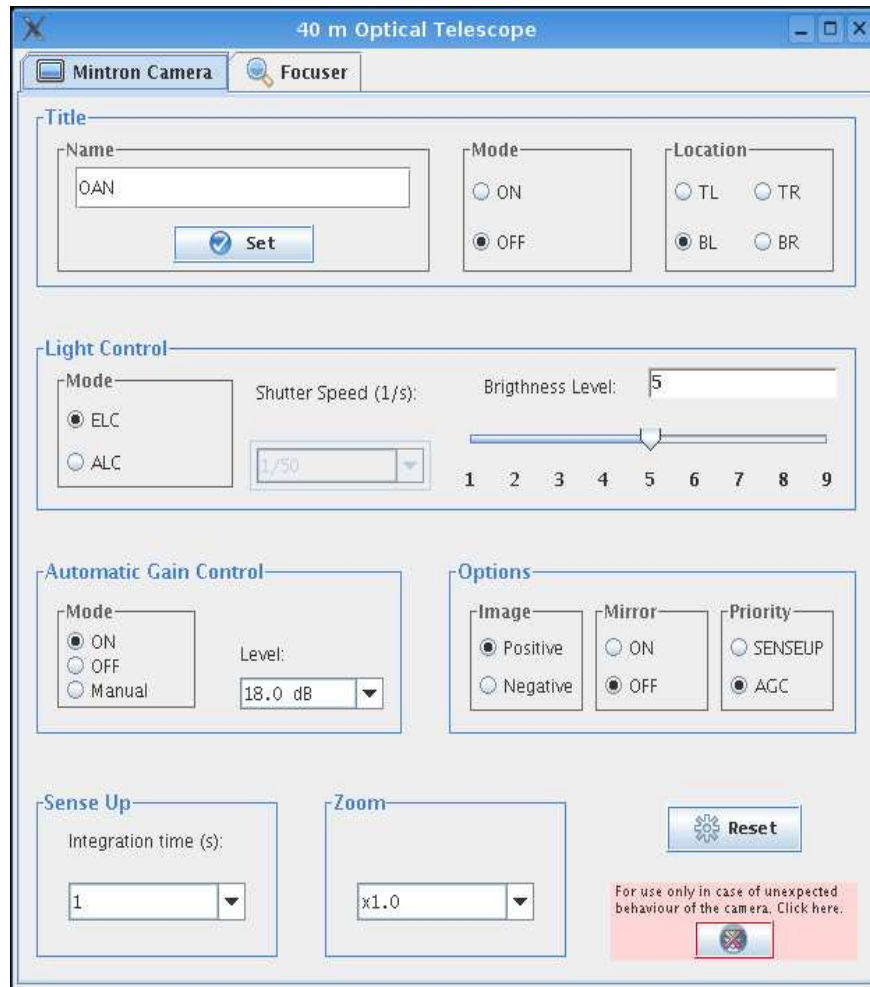
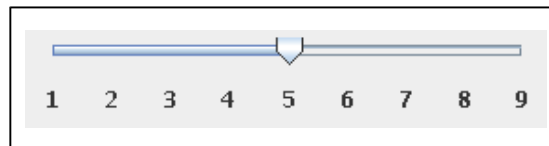


Fig. 14. Interfaz gráfica “Mintron camera”

Control de luz

- ✓ Seleccionar modo electrónico o automático -> *Mode ELC/ALC*.
- ✓ Ajustar el nivel de brillo -> Deslizador entre 1 y 9.



- ✓ Velocidad del obturador (sólo activado en modo ALC) -> Lista desplegable entre 1/50 y 1/12000

Control de ganancia automático

- ✓ Seleccionar automático, manual o desactivado -> *Mode ON/OFF/MANUAL*.
- ✓ Seleccionar el nivel de ganancia en dB --> *Level* entre 0 y 18 dB (sólo activado en los modos ON y MANUAL)

Opciones

- ✓ Imagen en positivo o negativo --> *Image Positive/Negative*
- ✓ Imagen especular--> *Mirror ON/OFF*
- ✓ Modo de funcionamiento de la cámara -> *Priority SENSEUP/AGC*

Sensibilidad

- ✓ Tiempo de integración--> *Integration time* lista desplegable entre 1 y 128 segundos.

Zoom

- ✓ Activar-->Lista desplegable entre x1.25 y x2.0
- ✓ Desactivar --> Lista desplegable x1.0

Reset

- ✓ Valores predefinidos en fábrica --> botón *Reset*

Si la cámara se queda en un estado inesperado es necesario reiniciarla a través de los botones de su panel trasero.

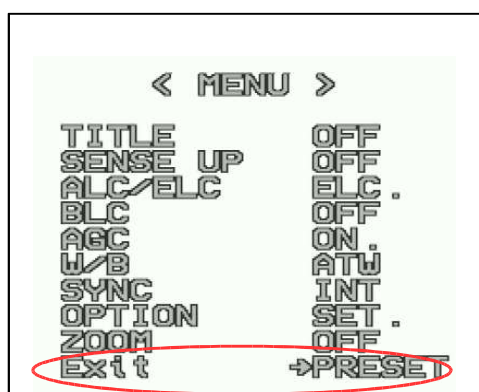
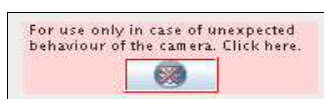


Fig. 15. Opción PRESET de la cámara

Como se ha comentado en secciones anteriores el sistema óptico de enfoque estará situado en el subreflector del radiotelescopio, un lugar de difícil acceso. Para simular los botones del panel trasero de la cámara se ha creado la interfaz gráfica “Rear Panel”. A esta interfaz se accede desde la interfaz gráfica principal.



Hay que tener en cuenta que SÓLO se debe utilizar esta interfaz en el caso de que la cámara Mintron no responda correctamente. Además una vez reseteada la cámara será necesario pulsar el botón *Reset* de la pestaña “Mintron Cámara” para que queden sincronizadas.

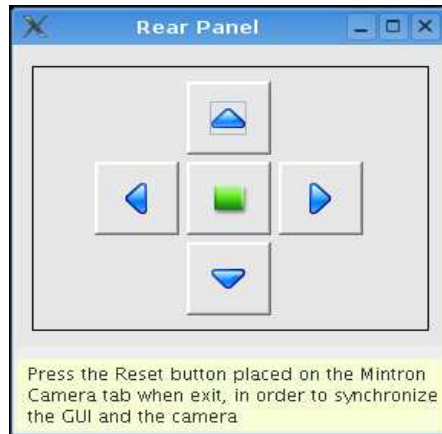


Fig. 16. Interfaz gráfica “Rear Panel”

El formato de la pestaña del enfocador TCF-S se muestra en la figura siguiente. Esta interfaz visualiza el estado del enfocador: temperatura, posición y pendientes °C/pasos de los modos automáticos (slopeA y slopeB). También permite seleccionar el modo de operación y comandar la posición y las pendientes A y B.

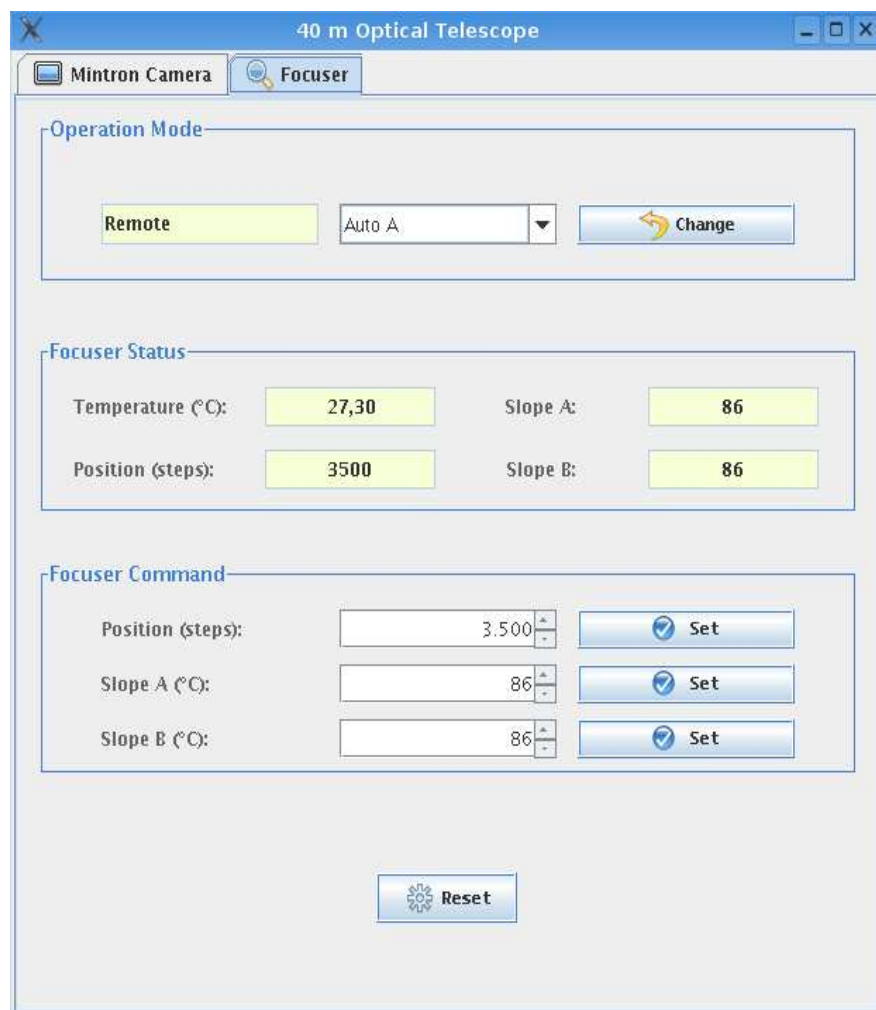


Fig. 17. Interfaz gráfica “Focuser”

Para cambiar el modo de operación se selecciona el modo deseado en la lista desplegable y se pulsa el botón *Change*. El comportamiento de la interfaz es distinto en cada uno de los casos:

Remoto

- ✓ Permite comandar la posición, slopeA y slopeB --> en los cuadros incrementales se indica el valor deseado y se pulsa el botón *Set*.
- ✓ Permite pasar a cualquiera de los otros modos de operación.

Automático A

- ✓ Únicamente visualiza el estado de la temperatura y la posición, actualizándolos continuamente.
- ✓ No permite comandar ningún valor.
- ✓ Sólo permite pasar a modo remoto.

Automático B

(Análogo al modo AutomáticoA)

Manual

- ✓ No visualiza el estado de ninguna variable.
- ✓ No permite comandar ningún valor.
- ✓ Sólo permite pasar a modo remoto.

Sleep

- ✓ No visualiza el estado de ninguna variable.
- ✓ No permite comandar ningún valor.
- ✓ Sólo permite pasar a a la opción *Wake Up*, que deja el enfocador en modo remoto.

El botón *Reset* pone el enfocador con los valores predefinidos y lo cambia a modo remoto.

2.4 Pruebas del sistema

Una vez desarrollado el software del sistema óptico de apuntado se ha realizado una batería de pruebas para comprobar su correcto funcionamiento y detectar posible errores.

2.4.1 Cámara CCD Mintron

Para realizar las pruebas de la cámara Mintron se ha utilizado una tarjeta gráfica modelo FlushBus Spectrim de la marca Integral Technologies. Esta tarjeta ha sido instalada en un PC Linux y se le ha conectado la salida de video compuesto de la

cámara. El driver de la tarjeta ofrece una aplicación denominada “offscreen” que permite visualizar en la pantalla del ordenador la imagen que recibe la tarjeta.

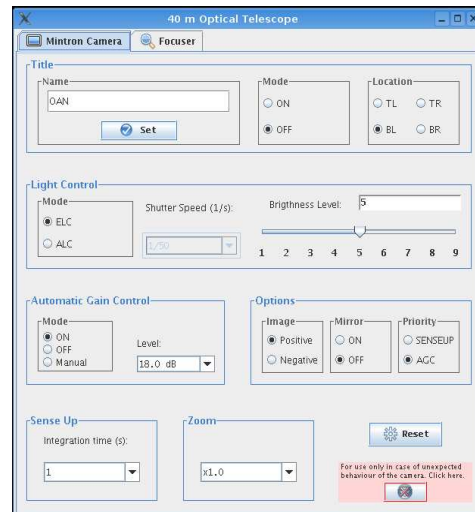
El telescopio óptico se ha orientado hacia una posición aleatoria a través de una ventana para obtener una imagen del paisaje.

Se ha comprobado el correcto funcionamiento de cada una de las opciones que ofrece el cliente gráfico, observando que los valores seleccionados en la interfaz gráfica se establecen en cada una de las propiedades definidas en el componente ACS, y por tanto en la configuración de la cámara. En primer lugar visualizando como cambian las opciones de configuración de la cámara a través del menú que aparece sobre impresionado en la imagen y en segundo lugar viendo el estado final de la imagen. A continuación se muestra el ejemplo de una de las pruebas en la que se cambia la imagen de positivo a negativo.

Imagen inicial



Opción seleccionada



Menú cámara

(justo antes de cambiar a NEGA)

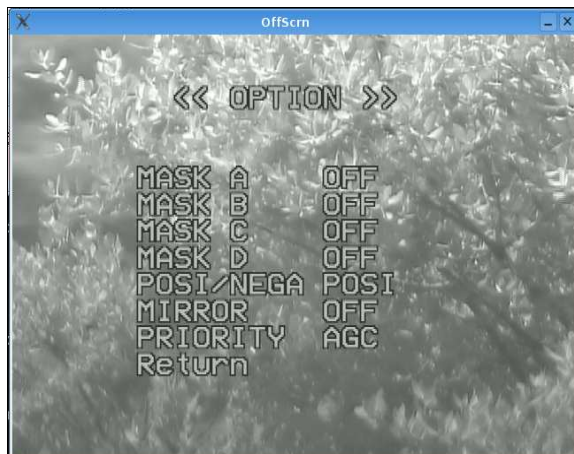


Imagen final



La selección de la posición del título, no funciona adecuadamente. Se ha podido comprobar que los comandos se envían a la cámara correctamente, como se pone de manifiesto por medio los loggers introducidos en el código. Sin embargo la cámara no ejecuta todos los comandos recibidos, y queda en un estado erróneo. El problema puede ser debido al microprocesador instalado en la cámara. Por este motivo para poder resetear la cámara sin tener que acceder al subreflector se ha creado la interfaz gráfica “Rear Panel” descrita en la sección 2.3.

2.4.2 EnfocadorTCFS

Se ha comprobado el correcto funcionamiento de cada una de las opciones que ofrece el cliente gráfico. Para todas las configuraciones los valores seleccionados en la interfaz gráfica se establecen en cada una de las propiedades definidas en el componente ACS, y por tanto en la configuración del enfocador. En primer lugar se puede ver como la posición va variando por medio del visor de la caja de control o por el sonido del motor al cambiar de posición. Del mismo modo en los modos automáticos se puede comprobar como cambia la posición de manera automática en el visor y en la interfaz gráfica al calentar el sensor de temperatura con la mano.

Al ir realizando pruebas se pudo comprobar que la comunicación entre el sistema de control remoto y los dispositivos no siempre era correcta. Tras analizar la causa del fallo se detectó que el problema era debido a la conexión entre el conversor Lantronix y los dispositivos. A través de una conexión al Lantronix por medio de su dirección IP se enviaron de modo remoto comandos a ambos dispositivos para comprobar si estos respondían adecuadamente. Únicamente respondían adecuadamente seleccionando el puerto adecuado del conversor para cada dispositivo.

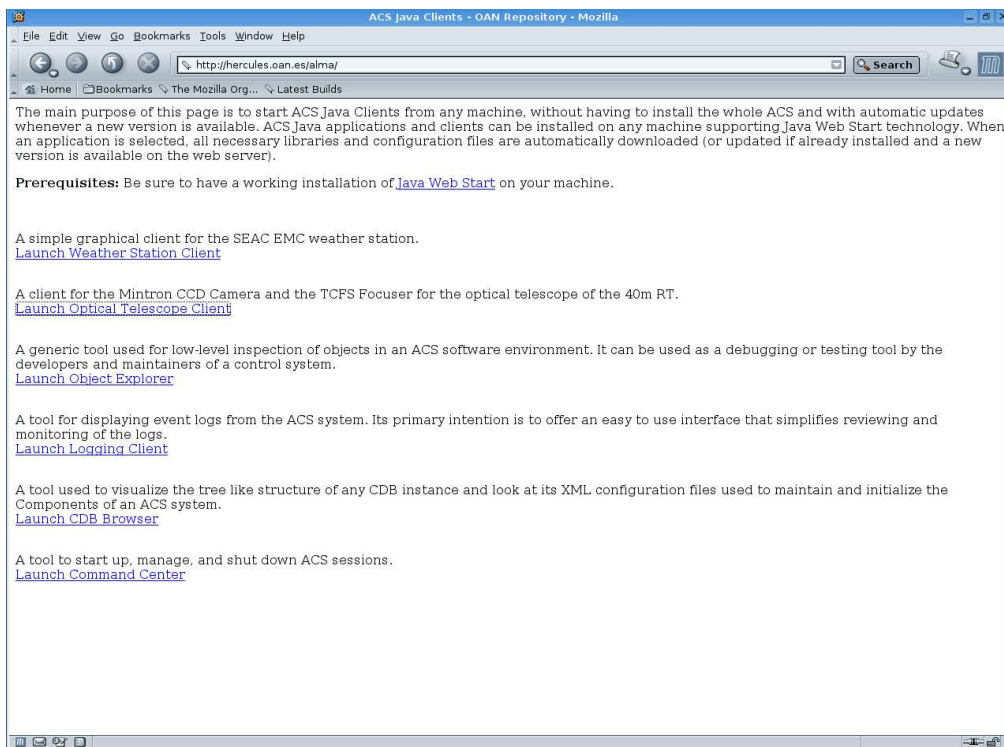
2.5 JavaWebStart

El software de Java Web Start desarrollado por Sun Microsystems permite descargar y ejecutar aplicaciones Java desde la Web, además permite activar las aplicaciones con un simple clic, garantiza que se está ejecutando la última versión de la aplicación y elimina complejos procedimientos de instalación o actualización. Este software permite arrancar las aplicaciones Java comprobando previamente si el cliente tiene la versión actualizada de dicha aplicación. Si no es así descarga la última versión y se ejecuta en local.

Actualmente Java Web Start viene incluido en el JRE (Java Runtime Environment) y ofrece tres métodos para ejecutar una aplicación:

- Desde enlaces en una página Web
- Desde el administrador de aplicaciones incorporado
- Desde un icono del escritorio y el menú Inicio (sólo en Microsoft Windows)

El sistema de control remoto del radiotelescopio de 40 m utiliza Java Web Start. Las aplicaciones desarrolladas en Java para el radiotelescopio se almacenan en un servidor web en el que reside una página web HTML con los enlaces a las distintas aplicaciones. Estos enlaces apuntan a un fichero .jnlp que indica la ruta de la aplicación en el servidor, la versión, etc. En este servidor web de aplicaciones se ha añadido el cliente Java del telescopio óptico de apuntado.



Para configurar el fichero .jnlp del telescopio óptico (OpticalTelClient.jnlp) se necesitan todos los archivos .jar que utiliza el cliente Java: socketError.jar, mintronError.jar, focuserError.jar, mintronCamera.jar, TCFSfocuser.jar y OpticalTelClient.jar. de este modo se permite tener acceso a ellos y se puede ejecutar la aplicación directamente.

3. Referencias

- [1] FROUFE, A. *JAVA 2: Manual de usuario y tutorial, 2ª Ed.* Madrid: Edit. RA-MA, 2000.
- [2] P. de Vicente, J.A. Lopez Perez, D. Cordobés, R. Bolaño, C. Almendros. *Robotización y control remoto de la camara Mintron.* Informe Técnico IT-OAN 2006-2.
- [3] P. de Vicente. *Control remoto del sistema de enfoque para el telescopio óptico del RT de 40m.* Informe Técnico IT-OAN 2006-5.
- [4] R. Bolaño. *Distribuyendo aplicaciones con Java Web Start .* Informe Técnico IT-OAN 2006-8.
- [5] JAVA Homepage: <http://java.sun.com/>
- [6] ACS Homepage: <http://www.eso.org/~gchiozzi/AlmaAcs/>