Joint Astronomy Centre
James Clerk Maxwell Telescope

PTCS TCS/UN/004
Nick Rees, Russell Kackley
4 August 2003

Description of the JAC Telescope Control System serial line interface

Contents

1 Introduction

Normally, when used for observing by JAC common-user instrumentation, the JAC telescope control systems are controlled through DRAMA task interfaces, either:

- The JCMT TEL interface (which is a backwards compatable interface which supports the modes of the previous TCS for old instrumentation).
- The native JCMT PTCS interface (PTCS/001).
- The OCS interface (OCS/ICD/006).

However, this setup neglects visitor instruments that normally do not support the DRAMA system. To circumvent this problem we provide a simple ASCII interface that can either be connected over a serial line, or to a well known TCP/IP port. This document describes this interface.

2 Interface description

The interface is supported in two forms:

- A portable 'C' library which can interface to a serial line on the instrument computer.
- The definition of the actual characters that run on a serial line or over a socket intterface.

With the exception of the C routine that initializes the serial line, there is a one to one correspodence between the C function calls and the serial line commands.

All the commands are described in section 7.

2.1 Description of the C library

The C library is provided as two C source files, plus associated header files.

The software can be obtained by anonymous ftp from the `pub/ukirt/ukirt_sw/telescope` directory on the machine `ftp.jach.hawaii.edu`.

The first source file (`siolib.c`) and its header file (`siolib.h`) contains just two routines to handle the serial line. These may vary from machine to machine since the serial interfaces may differ. One of the routines (`SioOpen`) opens the serial line and the other (`SioRead`) performs a two pass transaction on the line.

The second file (`tels.c`, header file `tels.h`) contains the telescope interface routines themselves. This should be portable since it just uses routines from the `stdio` library to encode and decode the routine parameters onto the character strings.

The calling sequence of the routines in the `tels` library are outlined in section 7.

2.2 Description of the serial line commands

From the visitor instrument's point of view, the serial line commands all consist of a command name followed by a fixed set of input parameters terminated by a carriage return (ASCII 13). The command and all the parameters are delimited by spaces. Each parameter is of one of four data types and every parameter must be present and in the correct position on the command line. A description of the data types that are used is given below.

Data type	Example	Description
char	'A string'	Text character string delimited by apostrophes (ASCII 39) and with no embedded apostropes.
double	1.45E+10	Character representation of a floating point number. A base 10 exponent can be included by prefixing the exponent with the letter 'E'.
integer	42	Character representation of a signed integer between $2^{31} - 1$ and $-(2^{31} - 1)$.
logical	TRUE	Text string with no embedded spaces. A first character of upper or lower case 't' indicates a true logical value. A first character of upper or lower case 'f' indicates a false value. Otherwise the result is an error. The telescope control system will return either 'TRUE' or 'FALSE' for these fields.

When the command has completed the telescope control system will return a response. The response consists of a set of returned parameters one of which, a status integer, is always present and is the first parameter in the response line. A returned status of zero indicates the command succeeded, anything else indicates some sort of failure. Note that in some cases (specifically the commands with the prefix `SET_`) command success only means that the command has been passed to the telescope control system, it does not indicate the command has been executed.

Note that the commands must be synchronous – i.e. two commands cannot be active at one time.

2.3 Example

In this example we want to do a number of things, which are fairly typical for a visitor instrument observing sequence:

1. Define a target position.
2. Slew to the target. When slew is finished telescope will be on target and ready to observe.
3. Offset 600 arcseconds North. When offset is finished telescope will be on the offset position and ready to observe.
4. Offset back to the centre position.

The commands for this are:

Command sent to telescope	Response
SET_TARGET 'NGC6251' 'B1950' 4.33772497 1.44322245 0 0 1950 0 0 0 0 0 'Galaxy' 0 0 0<CR>	0<CR>
SLEW 'MAIN'<CR>	0<CR>
OFFSET 1000 0<CR>	0<CR>
OFFSET 0 0<CR>	0<CR>

3 Status values

A (hopefully complete) list of the possible return status values is available as VMS .msg files, or as FORTRAN, Pascal or C header or include files. These files have a constant defined equivalent to every error message, and the definition has a text comment which describes the message.

There is also a small portable C library which is part of the DRAMA software which can take an error number and return the error string.

4 TCP/IP port pragmatics

At the current time, we have not determined the exact form of connection and port number to be used if the interface is to be a TCP/IP network port. This, however, will be agreed with the SMA project, who is likely to be the first project to use the interface in this way. It may be that we set up the port on a terminal server, and route this to a standard serial interface.

5 Serial line pragmatics

If the serial line interface is to be used, the serial line will be connected to a terminal server and so baud rate, parity, handshaking etc are fairly flexible. All characters on the line should be printable ASCII (except for the terminating carriage returns).

The default setup will be 9600 baud, 8 bits, no parity, XON/XOFF handshaking. The latter should not be required since the buffer sizes should be large enough to cope.

6 Command summaries

GET_AIRMASS**:** Get the airmass at the current telescope position

GET_DEMAND**:** Get the current telescope demand tracking coordinates

GET_GUIDING**:** Return autoguiding status

GET_IMAGE_SCALE**:** Return the image scale at the current focal plane

GET_OBSERVATORY**:** Get observatory information

`GET_OFFSETS`**:** Get offsets from base in gnomonic coordinates.

`GET_STATE`**:** Get the telescope `STATE` values

`GET_SYSTEM`**:** Get tracking coordinate system

`GET_TARGET`**:** Get the coordinates of a target object

`GET_TEL_BASE`**:** Get the virtual telescope base position

`GET_TIME`**:** Get various sorts of time

`NOD`**:** Nod telescope to selected chopper beam

`AOFFSET`**:** Offset the aperture from base

`OFFSET`**:** Offset the main and/or guide telescopes or the aperture from base

`TOFFSET`**:** Offset the main telescope from base

`XOFFSET`**:** Offset the guide telescope from base

`SET_BASE_HERE`**:** Set the virtual telescope base position to its current position

`SET_GUIDING`**:** Turn autoguiding on/off

`SET_TARGET`**:** Specify the coordinates of a target object

`SLEW`**:** Slew telescope to the next target

7 Detailed Command Descriptions

7.1 AOFFSET — Offset the aperture from base

Function: Offset the aperture from base

Description: Offset the aperture a specified amount from base. The aperture is offset in the focal plane.

Arguments: AOFFSET expects the following arguments to be supplied on the command line in the order shown.

1	OFFRA	double	The offset in the EW direction (arcsec)
2	OFFDEC	double	The offset in the NS direction (arcsec)

Returned Arguments: AOFFSET returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1	STATUS	Int	The status value

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002

7.2 GET_AIRMASS — Get the airmass at the current telescope position

Function: Get the airmass at the current telescope position

Description: Return the airmass at the current telescope position

Returned Arguments: GET_AIRMASS returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1	STATUS	Int	The status value
2	AIRMASS	Double	The airmass

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002

7.3 GET_DEMAND — Get the current telescope demand tracking coordinates

Function: Get the current telescope demand tracking coordinates

Description: Return the demand position of the main or guide telescope in the selected coordinate system.

Arguments: GET_DEMAND expects the following arguments to be supplied on the command line in the order shown.

1	GUIDE	char	TRUE for guide telescope, FALSE for main telescope.
2	SYSTEM	char	The coordinate system

TRACKING	The current tracking system
Jnnnn	FK5 mean for equinox nnnn
Bnnnn	FK4 mean for equinox nnnn
APP	Apparent RA/Dec
HADEC	Apparent Hour Angle/Dec
AZEL	Topocentric Azimuth/Elevation
MOUNT	Telescope Mount Coordinates

Returned Arguments: GET_DEMAND returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1	STATUS	Int	The status value
2	C1	Double	The RA or HA or Az in radians
3	C2	Double	The Dec or El in radians

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002

7.4 GET_GUIDING — Return autoguiding status

Function: Return autoguiding status

Description: Returns the actual autoguiding status (either on or off).

Returned Arguments: GET_GUIDING returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1	STATUS	Int	The status value
2	AUTO	char	TRUE if autoguiding is actually on

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002

7.5 GET_IMAGE_SCALE — Return the image scale at the current focal plane

Function: Return the image scale at the current focal plane

Description: Returns the image scale at the current focal plane. This can be used to relate the XY coordinate system to angles on the sky.

Returned Arguments: GET_IMAGE_SCALE returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1	STATUS	Int	The status value
2	SCALE	Double	The image scale (radians per mm)

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002

7.6 GET_OBSERVATORY — Get observatory information

Function: Get observatory information

Description: Returns the telescope name, longitude, latitude and height.

Returned Arguments: GET_OBSERVATORY returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1	STATUS	Int	The status value
2	NAME	char	The telescope name
3	LONG	double	East longitude - radians
4	LAT	double	Latitude - radians
5	HT	double	Height - metres

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002

7.7 GET_OFFSETS — Get offsets from base in gnomonic coordinates.

Function: Get offsets from base in gnomonic coordinates.

Description: Get the offset from base for either the demand position or the actual telescope position expressed as gnomonic (tangent plane) coordinates.

Arguments: GET_OFFSETS expects the following arguments to be supplied on the command line in the order shown.

1	GUIDE	char	TRUE for guide telescope, FALSE for main telescope.
2	DEMAND	char	TRUE for the demand position, FALSE for the actual telescope position.

Returned Arguments: GET_OFFSETS returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1	STATUS	Int	The status value
2	DC1	Double	Offset from base in EW direction - arcsec
3	DC2	Double	Offset from base in NS direction - arcsec

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002

7.8 GET_STATE — Get the telescope STATE values

Function: Get the telescope STATE values

Description: Return the state number, time, and telescope coordinates from the telescope STATE structure

Arguments: GET_STATE expects the following arguments to be supplied on the command line in the order shown.

1 TIME char The type of time value requested.

The following types are supported:

TAI	International Atomic Time (as MJD)
UTC	Coordinated Universal Time (as MJD)
UT1	Universal Time (as MJD)
TDB	Barycentric Dynamical Time (as MJD)
LAST	Local Apparent Sidereal Time (as fraction of day)

(2) SYSTEM (char) The coordinate system in which to express telescope coordinates. The following systems are supported:

TRACKING	The current tracking system
AZEL	Topocentric Azimuth/Elevation

Returned Arguments: GET_STATE returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1	STATUS	Int	The status value
2	NUMBER	Int	The state number
3	TIME	Double	The time value (as MJD or fraction of day, depending on the TIME input argument value)
4	AIRMASS	Double	The airmass
5	C1	Double	The first coordinate value (radians)
6	C2	Double	The second coordinate value (radians)

Support: Russell Kackley, JAC

Version Date: 18-May-2005

7.9 GET_SYSTEM — Get tracking coordinate system

Function: Get tracking coordinate system

Description: Get the name of the current tracking coordinate system for the main or guide telescope. This is the coordinate system that applies to the OFFSET, TOFFSET, and XOFFSET actions.

Arguments: GET_SYSTEM expects the following arguments to be supplied on the command line in the order shown.

1	GUIDE	char	TRUE for guide telescope, FALSE for main telescope.

Returned Arguments: GET_SYSTEM returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1	STATUS	Int	The status value
2	SYSTEM	char	The coordinate system

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002

7.10 GET_TARGET — Get the coordinates of a target object

Function: Get the coordinates of a target object

Description: The information on a target object is returned. Either the next or current target of the main telescope may be selected.

Arguments: GET_TARGET expects the following arguments to be supplied on the command line in the order shown.

1	NEXT	char	TRUE for NEXT target FALSE for CURRENT target

Returned Arguments: GET_TARGET returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1	STATUS	Int	The status value
2	NAME	char	The name of the target
3	SYSTEM	char	The coordinate system
4	C1	double	The RA or HA or Az in radians
5	C2	double	The Dec or El in radians
6	PMRA	double	For mean coordinates the RA proper motion
7	PMDEC	double	For mean coordinates the Dec proper motion
8	EPOCH	double	The epoch
9	PARALLAX	double	The parallax in seconds of arc
10	RV	double	The radial velocity in km/s
11	P1	double	First additional parameter
12	P2	double	Second additional parameter
13	P3	double	Third additional parameter
14	COMMENTS	char	The COMMENTS parameter
15	P4	double	Fourth additional parameter
16	P5	double	Fifth additional parameter
17	P6	double	Sixth additional parameter

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002

7.11 GET_TEL_BASE — Get the virtual telescope base position

Function: Get the virtual telescope base position

Description: Get the base position of the main or guide telescope. The position is returned in the current tracking coordinate system.

Arguments: GET_TEL_BASE expects the following arguments to be supplied on the command line in the order shown.

1	GUIDE	char	TRUE for guide telescope, FALSE for main telescope.

Returned Arguments: GET_TEL_BASE returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1	STATUS	Int	The status value
2	C1	Double	The RA or HA or Az in radians
3	C2	Double	The Dec or El in radians

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002

7.12 GET_TIME — Get various sorts of time

Function: Get various sorts of time

Description: Returns the Modified Julian Date (MJD), the UTC, UT1, Barycentric Dynamical Time (TDB) and Local Apparent Sidereal Time (LAST).

Returned Arguments: GET_TIME returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1	STATUS	Int	The status value
2	MJD	double	Modified Julian Date
3	UTC	double	UTC - days
4	UT1	double	UT1 - days
5	TDB	double	Barycentric Dynamical Time - days
6	LAST	double	Local Apparent sidereal Time - days

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002

7.13 NOD — Nod telescope to selected chopper beam

Function: Nod telescope to selected chopper beam

Description: The telescope is 'nodded' to the specified chopper beam. Beams of 'A' and 'B' will normally be set up to correspond to the two extremes of the chop throw, but other beams may be defined for special purposes. 'MIDDLE' normally corresponds to a point halfway between A and B. Completion of the nod is controlled by the tracking status of the main and guide telescopes.

Arguments: NOD expects the following arguments to be supplied on the command line in the order shown.

1	BEAM	char	The name of the required chopper beam

Returned Arguments: NOD returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1	STATUS	Int	The status value

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002

7.14 OFFSET — Offset the main and/or guide telescopes or the aperture from base

Function: Offset the main and/or guide telescopes or the aperture from base

Description: Offset the main and/or guide telescopes a specified amount from base. The offsets are specified in a gnomonic (tangent plane) projection with the current base position as the tangent point. Or the aperture is offset in the focal plane

Arguments: OFFSET expects the following arguments to be supplied on the command line in the order shown.

1	OFFRA	double	The offset in the EW direction (arcsec)
2	OFFDEC	double	The offset in the NS direction (arcsec)

Returned Arguments: OFFSET returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1	STATUS	Int	The status value

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002

7.15 SET_BASE_HERE — Set the virtual telescope base position to its current position

Function: Set the virtual telescope base position to its current position

Description: Set the base position of the main or guide telescope to its current demand position. The offsets are also set to zero, so the telescope will not move as a result of this operation.

Arguments: SET_BASE_HERE expects the following arguments to be supplied on the command line in the order shown.

1	GUIDE	char	TRUE for guide telescope, FALSE for main telescope.

Returned Arguments: SET_BASE_HERE returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1	STATUS	Int	The status value

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002

7.16 SET_GUIDING — Turn autoguiding on/off

Function: Turn autoguiding on/off

Description: This routine is used to specify whether autoguiding is on, in which case the telescope guiding corrections are obtained from the autoguider, or off, in which case no guiding corrections are applied.

Arguments: SET_GUIDING expects the following arguments to be supplied on the command line in the order shown.

1 AUTO char TRUE to turn autoguiding on

Returned Arguments: SET_GUIDING returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1 STATUS Int The status value

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002

7.17 SET_TARGET — Specify the coordinates of a target object

Function: Specify the coordinates of a target object

Description: The coordinates of a target are specified and the target becomes the NEXT target - i.e. the target the telescope will slew to when the slew action is requested. The target may be specified in a number of coordinate systems. For mean coordinates the proper motions, epoch, parallax and radial velocity must be specified. For other coordinate systems they are ignored. The comments and associated parameters are not used by the telescope control software, but are stored as a means of passing information between various software systems (e.g. the magnitudes of a standard from a catalogue searching task to an instrument task).

Arguments: SET_TARGET expects the following arguments to be supplied on the command line in the order shown.

1 NAME char The name of the target

2 SYSTEM char The coordinate system

Jnnnn	FK5 mean for equinox nnnn
Bnnnn	FK4 mean for equinox nnnn
APP	Apparent RA/Dec
HADEC	Apparent Hour Angle/Dec
AZEL	Topocentric Azimuth/Elevation
MOUNT	Telescope Mount Coordinates

3	C1	Double	The RA (or HA or AZ) in Radians
4	C2	Double	The Dec (or EL) in Radians
5	PMRA	Double	For mean coordinates - the RA proper motion in Radians per year
6	PMDEC	Double	For mean coordinates - the Dec proper motion in Radians per year
7	EPOCH	Double	For mean coordinates - the epoch in years AD
8	PARALLAX	Double	For mean coordinates - the parallax in seconds of arc
9	RV	Double	For mean coordinates - the radial velocity in km/s
10	P1	Double	Additional information associated with the target
11	P2	Double	Additional information associated with the target
12	P3	Double	Additional information associated with the target
13	COMMENTS	char	Comments to be associated with the target
14	P4	Double	Additional information associated with the target
15	P5	Double	Additional information associated with the target
16	P6	Double	Additional information associated with the target

Returned Arguments: SET_TARGET returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1	STATUS	Int	The status value

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002

7.18 SLEW — Slew telescope to the next target

Function: Slew telescope to the next target

Description: The SLEW action results in the following sequence of events.

1. The specified target information is copied to the CURRENT_TARGET.
2. The target position (corrected for proper motion and parallax if necessary) becomes the new telescope base position.
3. The offset from base is set to zero.
4. The differential tracking rates are set to those specified for the new source, or to zero if none were specified.
5. The SLEW action completes immediately. Use the WAIT action if you wish to wait until the telescope is tracking the source.

The action will return with an error if the next target is not currently accessible.

The OPTION and CYCLE arguments are only used in cases where there are multiple possible mount positions at which the source could be reached. The TRACK_TIME option is normally the recommended one to use.

Arguments: SLEW expects the following arguments to be supplied on the command line in the order shown.

1	VT	char	The name of the virtual telescope this operation applies to (e.g. MAIN or GUIDE). If not specified, MAIN is assumed. If the name is specified as ALL, the action will be applied to all existing virtual telescopes.
2	TARGET	char	The name of the target to use as the next target (e.g. NEXT, MAIN, GUIDE, etc.). If not specified, NEXT is assumed. However, if VT=ALL and TARGET is not specified, then each telescope will use a target with the same name as the telescope (e.g., the MAIN telescope will use the MAIN target, the GUIDE telescope will use the GUIDE target, etc.)
3	OPTION	char	The method for resolving multiple accessibility options for the source. SHORTEST_SLEW is the default if no option is specified. OPTION is one of:

SHORTEST_SLEW	Choose the option requiring the shortest slew
LONGEST_TRACK	Choose the option which gives the longest track time
TRACK_TIME	Choose the option which gives the shortest slew, whilst still maintaining, if possible, a minimum track time of the value specified in the TRACK_TIME parameter.
CYCLE	Use explicitly specified cycle and/or sector

If OPTION is CYCLE, supply the following additional component.

4 CYCLE Double Cycle number in Azimuth-like axis to check. Every other half cycle the elevation will be > 90 degrees. Cycles are:

0	means mount azimuth or HA in range 0 to 180 degrees,
1	in range 360 to 540 degrees,
-1	in range -360 to -180 degrees etc.

If OPTION is TRACK_TIME, supply the following additional component.

4 TRACK_TIME Double Minimum track time (in days) for the TRACK_TIME option.

Support: Russell Kackley, JAC

Version Date: 4-Aug-2003

7.19 TOFFSET — Offset the main telescope from base

Function: Offset the main telescope from base

Description: Offset the main guide telescope a specified amount from base. The offsets are specified in a gnomonic (tangent plane) projection with the current base position as the tangent point.

Arguments: OFFSET expects the following arguments to be supplied on the command line in the order shown.

1 OFFRA double The offset in the EW direction (arcsec)

2 OFFDEC double The offset in the NS direction (arcsec)

Returned Arguments: TOFFSET returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1 STATUS Int The status value

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002

7.20 XOFFSET — Offset the guide telescope from base

Function: Offset the guide telescope from base

Description: Offset the guide telescopes a specified amount from base. The offsets are specified in a gnomonic (tangent plane) projection with the current base position as the tangent point.

Arguments: XOFFSET expects the following arguments to be supplied on the command line in the order shown.

1 OFFRA double The offset in the EW direction (arcsec)

2 OFFDEC double The offset in the NS direction (arcsec)

Returned Arguments: XOFFSET returns a space-delimited string containing the following quantities in the order shown. The string is terminated with a <CR>.

1 STATUS Int The status value

Support: Russell Kackley, JAC

Version Date: 4-Apr-2002
