THE OBSERVATORIES OF THE CARNEGIE INSTITUTION OF WASHINGTON
LAS CAMPANAS OBSERVATORY

Colina El Pino s/n
La Serena, Casilla 601
Phone: 56-51-207301 • Fax: 56-51-207308

TERTIARY MIRROR CONTROL SYSTEM SOFTWARE

User Manual

Document Code	**01-00-J4**
Type	User Manual
Author	Silvia Baeza and Jose M. Soto
Date	April, 2008
Revision	Jose M. Soto

Revisions Table

Date	Description
April, 2008	Document Revision
March 13, 2008	Document Revision
April 23, 2003	Document Created

Table of Contents

Overview

1

1 Overview of the Tertiary Mirror Control System

1.1 General Configuration

The tertiary mirror is installed into a compact set of two turrets. This assembly rests mounted in the center of the primary mirror right above it. One of the turrets (the inner one) is called M3 turret and supports the tertiary mirror in its cell. The outer one, called ADC turret, holds the two Atmospheric Dispersion Compensator lenses (ADC lenses).

The main purpose of the whole system is to deviate the light collected by the telescope, to one of the six possible observing ports: two Nasmyth: east or west; three Folded ports: AUX1, AUX2 or AUX3, and Cassegrain. In the last case, the tertiary mirror doesn't deviate the light, instead, it just moves up out of the way. This task is achieved rotating the M3 turret.

Other important feature, is to compensate for the atmospheric dispersion, by using the corrector lenses. This can by accomplished rotating the ADC turret so the ADC lenses be in the light path; and then making the mirrors turn in concentric but opposite directions.

Both turrets and both ADC lenses are driven by stepper motors, the first two directly coupled to the rotating assembly, and the second ones coupled to the lenses through a chain.

The M3 turret and the ADC turret movements are independent from each other, so each of the six ports that can be pointed at, may or may not use the ADC corrector, giving a total number of 12 possible observing configurations.

There are three more stepper motors for tip-tilting the tertiary mirror with respect to its cell and three position sensors (LVDT's) to measure the position in each point.

Finally, there are 3 load cells that measure the M3 weight against the cell. Between the mirror and the cell there is a sealed chamber that contains air and supports the tertiary weight. The air can be injected to or extracted from the chamber, using two small selenoid valves connected to the main air and vacuum supply lines of the telescope, respectively.

The software is required to perform a number of operations:

- Accept high level commands to position the M3 mirror, ADC and other devices, and convert them to low level commands for the various motor controllers

- Report the position and status of the tertiary mirror and devices.

- Allow tertiary mirror motion. The mirror motion will allow to deviate the light through one of 6 different observing ports.

- Allow ADC turret motion, choosing ADC or no ADC position.

- Allow pressure/ vacuum control for mirror support.

1.2 Devices

There are seven Mycom stepper drives to command the following devices: Tertiary mirror turret, ADC turret, ADC A position, ADC B position, tilt A, tilt B and tilt C.

There is a computer coupled to the tertiary mirror turret. It talks with the stepper drives and processes information obtained from all sensors. This machine has three PC104 cards:

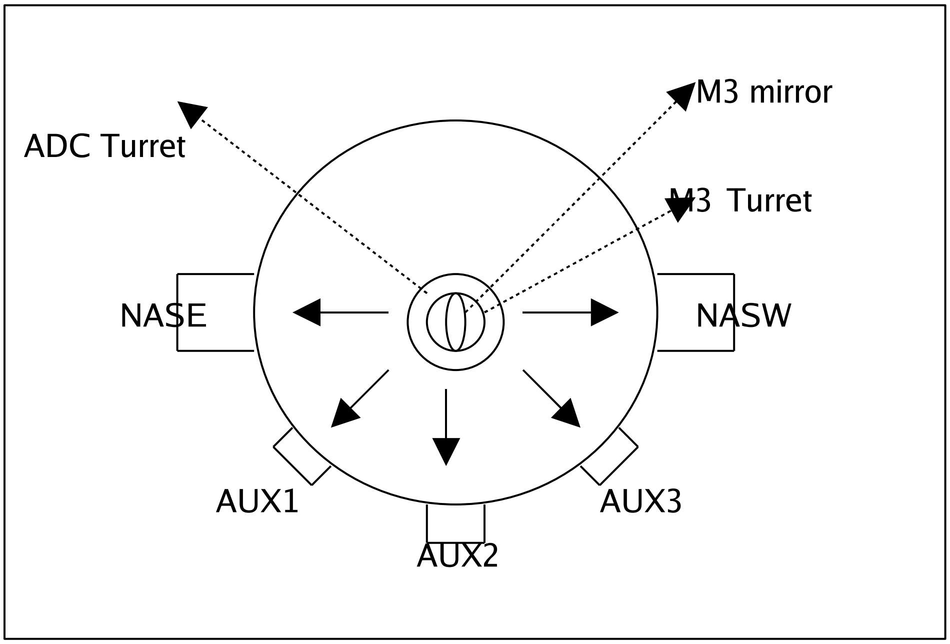
The CPU (MSM 586SEV) with a 386 type processor, running at 133 Mhz.

The digital IO card, model PCM-UIO48A

The MESA card, which is a 16 bit resolution digital to analog converter.

1.3 Positions

The following drawing (top down view) shows the positions of the observing ports in the telescope:



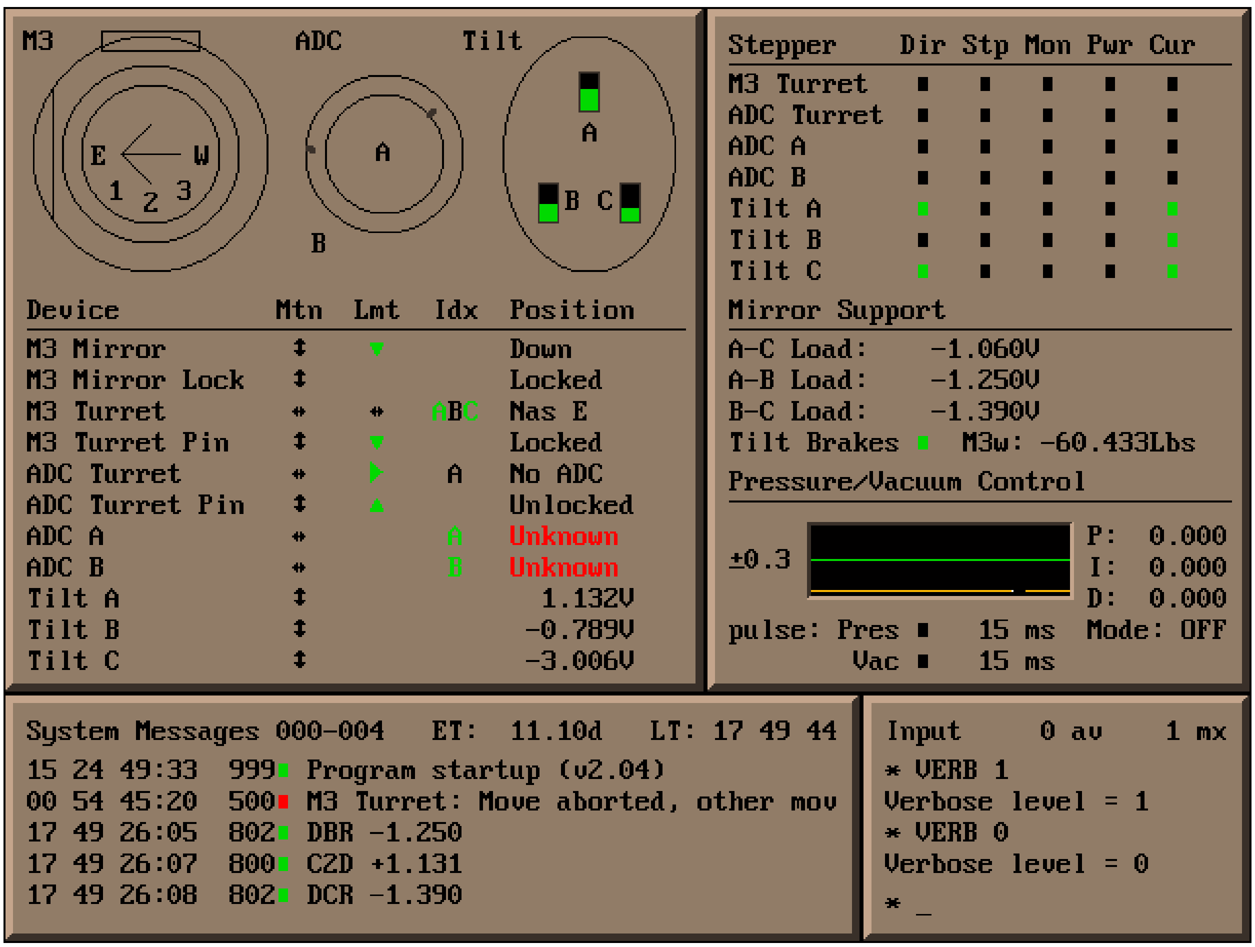
The M3 turret rotates to point the tertiary mirror to any of the six observing ports. In every position, the ADC turret can be in the ADC or no ADC position.

Interface

2

2 Tertiary Mirror Control System Interface

The following picture shows the M3 Control system interface:



The interface is divided in four sections.

The upper left part, shows the tertiary turret information: mirror position, M3 / ADC turrets positions, M3 / ADC pin status (Locked or unlocked), ADC lenses positions, and tilt readings.

The upper right side show the seven stepper motors status, together with mirror support information: load cell readings, and pressure/vacuum control figures.

The bottom area shows system log messages to the left, and the input command area.

2.1 Tertiary Turret Interface

2.2 Stepper Motors Interface

2.3 Mirror Support Interface

2.4 Pressure/vacuum Control Interface

The right window shows the Pressure/Vacuum Control. P, I and D labels, show the PID servo loop control numbers. You can toggle the servo on or off typing PVM. When you type it for the first time, the status "Mode" changes to AUTO and the Pressure (Pres) and Vacuum (vac) status flags on the right, turn green if the corresponding valve is opened. If you type PVM again, "Mode" changes to off, and the display shows the pres/vac valves closed (black).

Commands

3

3 Commands Summary

All M3 commands begin with a command code, followed by zero, one or more arguments. The arguments and the command must all be separated by spaces. Letters may be upper or lower case. The backspace key moves the cursor back one character at a time. All commands are terminated by a <cr>. Hit <F1> for a quick command help screen.

3.1 User Commands

Command	**Description**
SP	Toggle power to Tilts, ADC Turret and M3 Turret steppers
ASP	Toggle power to ADC Rotators steppers
M <*cmd*>	Mirror motion. <*cmd*> = UP / DOWN / STOP
MT <*cmd*>	Mirror Turret motion. <*cmd*> = HOME / STOP / AUX1 / AUX2 / AUX3 / NASW / NASE / LOCK / UNLOCK / <steps>
AT <cmd>	ADC Turret motion. <*cmd*> = HOME / ADC / NOADC / LOCK / UNLOCK / <steps>
ADCPOWER <*cmd*>	ADC lenses stepper power. <cmd> = ON / OFF. Not yet implemented
ADCA / ADCB	ADC Rotator A/B independent motion. <*cmd*> = HOME / STOP /

<cmd>	S<steps> / <degrees>
ADC *<cmd>*	ADC lenses combined operation. *<cmd>* = MAN / AUTO / <degrees>
TA / TB / TC *n*	Move Tilt A / B / C stepper *n* steps
PISTON *n*	Compound motion. Move Tilt A/B/C *n* millimeters
TH *n*	Compound horizontal motion. Move Tilt A *n*, Tilt B/C -*n* arcsecs
TV *n*	Compound vertical motion. Move Tilt B *n* arcsecs, Tilt C -*n* arcsecs
LAT *<cmd>*	Latches manual control. <cmd> = OPN / CLS
NASW	Rotate Mirror and ADC turret to Nasmyth West platform, adjusting the tilts
NASE	Rotate Mirror and ADC turret to Nasmyth East platform, adjusting the tilts
AUX1	Rotate Mirror and ADC turret to Aux 1, adjusting the tilts
AUX2	Rotate Mirror and ADC turret to Aux 2, adjusting the tilts
AUX3	Rotate Mirror and ADC turret to Aux 3, adjusting the tilts
TAS / TBS / TCS *<cmd>*	Servo individual Tilt A/B/C to set point. *<cmd>* = LVDT voltage
TILTS *<cmd1>* *<cmd2>* *<cmd3>*	Servo all three Tilts. *<cmd1>* = LVDT A voltage set point, *<cmd2>* = LVDT B voltage set point, *<cmd3>* = LVDT C voltage set point
TENASW *<ele>*	Adjust M3 Tilts according to elevation value. *<ele>* = 20, 25, 30, 35, ..., 90
PVS *<cmd>*	Change pressure/vacuum control chart scale
PVM	Toggle pressure/vacuum control on or off

PRESOFF	Close Pressure valve
PRESON	Open Pressure valve
VACOFF	Close Vacuum valve
VACON	Open Vacuum valve
TBRKOFF	Tilt brakes disengaged
TBRKON	Tilt brakes engaged

3.2 Message Commands

Command	Description
<Up>	Scroll system message display back one line
<Pg Up>	Scroll system message display back one page
<Down>	Scroll system message display forward one line
<Pg Dn>	Scroll system message display forward one page
<Home>	Set system message display to show the first message
<End>	Set system message display to show the current message

3.3 Display Commands

Command	Description
<F1>	Display online help screen
<Scroll Lock>	Pause/Resume system log messages
VERB *n*	Set system log verbose level to *n* (*n* = 0 ..2)

3.4 Engineering Commands

Command	Description
T1	Open a terminal on COM 1: at 19.2k (TCS)
T2	Open a terminal on COM 2: at 19.2k with DGH modules
T3	Open a terminal on COM 2: at 300 baud for DGH modules set up
FM	Force mirror up, even in unsafe position ! Use with extreme caution
<F2>	Dump system log to the file “TEMP.LOG”
<F3>	Dump screen to a file called “SCREEN.BMP” (in Windows BMP format)

3.5 Other Commands

Command	Description
Q / EXIT	Exit the program.

Configuration

4

4 M3 Turret Configuration

4.1 M3.ini file, Magellan I

```
; M3.INI
; 2/17/2003
;
; This file contains all easily modified initialization data for the
; M3 program. Lines starting with a ";" are ignored.
; Every other line contains a description followed by an equal sign
; followed by a data value. DO NOT change the order of the data values
; (the program ignores the description in front of them) and DO NOT
; add blank lines between values.
;
; System Parallel port address (in hex)
PARPORT = 378
; Serial communication constants
;  Address of this computer on the line to the host computer
Address = "J"
;  Prompt character for this computer
Prompt = ":"
;  Response character of this computer
```

```
Response = "~"

; Amount in parts-per-million to speed up or slow down the system clock

; Use this to make the system clock as accurate as desired

ClockPPM = 0

;

; Time in seconds before mirror motion times out

Mirror.MotionTimeout = 60

; Time in seconds for mirror latches to completely open/close

Mirror.FullLatchMove = 8

;

; M3 Turret acceleration in steps/sec/sec

M3Turret.Acceleration = 500.0

; M3 Turret maximum velocity in steps/sec

M3Turret.MaxVelocity = 500.0

; M3 Turret motion initial stepper pulse in milliseconds

M3Turret.InitialPulse = 50

; M3 Turret emergency stop distance (steps)

M3Turret.EmergencyStopDistance = 150

;

;  Time in seconds before M3 Turret Lock Pin times out

M3TurretPin.MotionTimeout = 30

;

; ADC Turret acceleration in steps/sec/sec

ADCTurret.Acceleration = 500.0

; ADC Turret maximum velocity in steps/sec

ADCTurret.MaxVelocity = 500.0

; ADC Turret motion initial stepper pulse in milliseconds

ADCTurret.InitialPulse = 50
```

```
; ADC Turret emergency stop distance (steps)

ADCTurret.EmergencyStopDistance = 150

;

; ADC Time in seconds before ADC Turret Lock Pin times out

ADCTurretPin.MotionTimeout = 30

;

; ADC A acceleration in steps/sec/sec

ADCA.Acceleration = 250.0

; ADC A maximum velocity in steps/sec

ADCA.MaxVelocity = 250.0

; ADC A motion initial stepper pulse in milliseconds

ADCA.InitialPulse = 50

; ADC A Steps per revolution

ADCA.StepsPerRevolution = 19200

;

; ADC B acceleration in steps/sec/sec

ADCB.Acceleration = 250.0

; ADC B maximum velocity in steps/sec

ADCB.MaxVelocity = 250.0

; ADC B initial stepper pulse in milliseconds

ADCB.InitialPulse = 50

; ADC B Steps per revolution

ADCB.StepsPerRevolution = 19200

;

; Tilt constants

;

; Tilt Stepper acceleration in steps/sec/sec

Tilt.Acceleration = 100.0
```

```
; Tilt Stepper maximum velocity in steps/sec

Tilt.MaxVelocity = 50.0

; Tilt Stepper initial stepper pulse in milliseconds

Tilt.InitialPulse = 50

; Tilt adjusts for observing ports in volts

; NASW

Tilt.TNASW[0] (NASW TA) =  1.22

Tilt.TNASW[1] (NASW TB) = -0.92

Tilt.TNASW[2] (NASW TC) = -2.92

; NASE

Tilt.TNASE[0] (NASE TA) = 1.11

Tilt.TNASE[1] (NASE TB) = -0.79

Tilt.TNASE[2] (NASE TC) = -3.09

; AUX1

Tilt.TAUX1[0] (AUX1 TA) = 0.0

Tilt.TAUX1[1] (AUX1 TB) = 0.0

Tilt.TAUX1[2] (AUX1 TC) = 0.0

; AUX2

Tilt.TAUX2[0] (AUX2 TA) = 0.0

Tilt.TAUX2[1] (AUX2 TB) = 0.0

Tilt.TAUX2[2] (AUX2 TC) = 0.0

; AUX3

Tilt.TAUX3[0] (AUX2 TA) = 0.0

Tilt.TAUX3[1] (AUX2 TB) = 0.0

Tilt.TAUX3[2] (AUX2 TC) = 0.0

;

; Pressure/Vacuum control constants

;
```

```
; Kp, Proportional PID control loop constant (~1-5)

PresVac.Kp = 0.7

; Ki, Integral PID control loop constant (~0.01-0.1)

PresVac.Ki = 0.004

; Kd, Derivative PID control loop constant (~1)

PresVac.Kd = 0.8

; PIl, Pressure Integral Limit (0.0-1.0)

PresVac.PIl = 0.2

; VIl, Vacuum Integral Limit (0.0-1.0)

PresVac.VIl = 0.2

; Integral Reset Threshold (~0.3-1.0)

PresVac.IntegralThreshold = 1.0

; Minimum pulse in ms for pressure valve

PresVac.PressureMinPulseWidth = 15

; Minimum pulse in ms for vacuum valve

PresVac.VacuumMinPulseWidth = 15

; Cycle time between control variable updates in ms

PulseCycle = 200
```

4.2 M3.ini , Magellan II

4.3 M3.ini Description

The M3 Utility program reads a small amount of configuration data from the file "M3.INI", which is in the same directory as the M3 Turret program. This is a simple text file which can be edited with any text editor. All lines starting with a ";" are ignored, all other lines are significant. Non-comment lines will have a label, an equal sign, and a value. Do not change the order of the lines, or add blank lines that aren't begun with a ";", as this will interfere with the M3 program's ability to read the file. The following values can be set in this file:

- PARPORT: System Parallel port address (in hex).
- Address: The address character that the M3 program responds to (as per the control system serial communication documents). This should be "J".
- Prompt: The prompt character that the M3 program responds to. This should be ":".
- Response: The response character that precedes all M3 replies. This should be "~".
- ClockPPM: A number, in parts per millionth of a second, to speed up or slow down the system clock. This should be used to match UT as closely as possible. To set, allow the TCS to set the M3's UT several times over the course of many hours. If the timestamps of the UT Set messages in the system log are ahead of the time being set, increase ClockPPM by 1000000 * (amount ahead by in seconds / time since last UT Set message in seconds), otherwise decrease by the same amount. This number will have to be modified if the computer being used to run the M3 program is changed. Set this number to zero if you're unsure.
- Mirror.MotionTimeout: Time in seconds before mirror motion times out.
- M3Turret.Acceleration: M3 Turret acceleration in steps/sec/sec.
- M3Turret.MaxVelocity: M3 Turret maximum velocity in steps/sec.
- M3Turret.InitialPulse: M3 Turret motion initial stepper pulse in milliseconds.
- M3Turret.EmergencyStopDistance: M3 Turret emergency stop distance (steps).
- M3TurretPin.MotionTimeout: Time in seconds before M3 Turret Lock Pin times out.
- ADCTurret.Acceleration: ADC Turret acceleration in steps/sec/sec.
- ADCTurret.MaxVelocity: ADC Turret maximum velocity in steps/sec.
- ADCTurret.InitialPulse: ADC Turret motion initial stepper pulse in milliseconds.
- ADCTurret.EmergencyStopDistance: ADC Turret emergency stop distance (steps).
- ADCTurretPin.MotionTimeout: ADC time in seconds before ADC Turret Lock Pin times out.
- ADCA.Acceleration: ADC A acceleration in steps/sec/sec.
- ADCA.MaxVelocity: ADC A maximum velocity in steps/sec.
- ADCA.InitialPulse: ADC A motion initial stepper pulse in milliseconds.
- ADCA.StepsPerRevolution: ADC A Steps per revolution.

- ADCB.Acceleration: ADC B acceleration in steps/sec/sec.
- ADCB.MaxVelocity: ADC B maximum velocity in steps/sec.
- ADCB.InitialPulse: ADC B initial stepper pulse in milliseconds.
- ADCB.StepsPerRevolution: ADC B Steps per revolution.
- Tilt.Acceleration: Tilt Stepper acceleration in steps/sec/sec.
- Tilt.MaxVelocity: Tilt Stepper maximum velocity in steps/sec.
- Tilt.InitialPulse: Tilt Stepper initial stepper pulse in milliseconds.
- Tilts adjust for observing ports in volts

Pressure/Vacuum control constants

- PresVac.Kp: Kp, Proportional PID control loop constant (~1-5).
- PresVac.Ki: Ki, Integral PID control loop constant (~0.01-0.1).
- PresVac.Kd: Kd, Derivative PID control loop constant (~1).
- PresVac.Pil: PIl, Pressure Integral Limit (0.0-1.0).
- PresVac.Vil: VIl, Vacuum Integral Limit (0.0-1.0).
- PresVac.IntegralThreshold: Integral Reset Threshold (~0.3-1.0).
- PresVac.PressureMinPulseWidth: Minimum pulse in ms for pressure valve.
- PresVac.VacuumMinPulseWidth: Minimum pulse in ms for vacuum valve.
- PulseCycle: Cycle time between control variable updates in ms.

4.4 Starting the Program

To start the M3 program, push the power button (the red switch) on the control box inside the M3 Turret. The program will load automatically from the solid-state disk. Resetting the computer insures that the system will start each time in the proper state. The initialization process consists of loading some MESA's drivers (running 4A22DRVR 220 9 61 followed by 4A22DCFG 61), and then the M3 program.

When the program is loaded, you should see the usual video display of M3 utility data. You can type commands which appear at the "*" prompt in the input panel. At the top of the input panel the average and maximum cycle times for the program's main loop are displayed in milliseconds.

Status messages are displayed in the system message box. The UT at which the message occurred is followed by a three-digit message identification code. A red LED indicates that the message describes error. A green LED indicates that the message describes a normal condition or success. Most errors are caused by serial communication errors, or by some piece of remote equipment being turned off. A certain number of errors may occur when the program is loaded, before the serial communications are properly initialized. A few errors will occur during normal operation.

The messages are numbered from 0 to 999. When the message buffer is full, new messages are written over the old ones starting from the beginning (so message 1000 is written to location 0 and so on). The most recent message is displayed in white. You can scroll back and forth through the message buffer using the cursor keys. To return to the page with the most recent message, be sure to use <End>, or else new messages will not appear on your screen (although new messages will *always* be written to the message buffer).

The elapsed time, in days, that the program has been running is shown next to the "ET" entry.

Serial Communication

5

5 Control System Serial Communication Standards (Not implemented yet)

There is always a host (upstream) computer, and a guest (downstream) computer. The host computer is frequently the TCS, with the guest being the tertiary mirror computer. The TCS system sends a command, and the guest responds immediately to that command. Guest computer never broadcast without being queried, which allows multiple guests to be chained on the same serial line.

Full command format:

```
:Jnddddddddddccr
```

The “ **:**” is the prompt character to initiate communication. **J** is the guest tertiary mirror computer’s unit address. **n** is a command number, **d**’s are data specific to the command (variable length), **cc** is a checksum, and **r** is a carriage return (ASCII 13).

Full response format:

```
~Jnddddddddccr
```

The “ **~**” is the response character for guest computer responses. **J** is the tertiary mirror computer’s unit address. **n** is the command number this is in response to, **d**’s are data specific to the response (variable length), **cc** is a checksum, and **r** is a carriage return (ASCII 13).

For very short commands and responses the checksum may be omitted (this is noted in the command description).

For all commands and responses that include a checksum, the checksum is composed of two hexadecimal digits (from 0-F). The checksum is calculated by starting with zero and XORing it with all characters in the message from the unit letter to the last data character before the checksum (the underlined part of the command and response above).

Commands that are received but misunderstood (checksum wrong, unknown command, etc) are replied to like this:

```
~J?r
```

Most guest computers maintain a running system log that contains important messages and all system status information. Each also maintains a pointer into that log that keeps track of the oldest message that hasn't been sent to the host computer. The "2" and "3" commands let the host computer command the guest computer to transmit one of its log entries or re-transmit the last entry. This is referred to as the "Engineering Data Stream", or EDS.

Command Summary:

2: Query Next EDS Message

3: Repeat Last EDS Message

4: Set UT

9: Free-form Command

5.1 Command Description

2: Query Next EDS Message

Commands the guest computer to send its oldest un-sent EDS log entry, and advance its internal pointer to the next EDS log entry.

Command Format: `:J2r` (note that this command has no checksum)

Response Format: `~J2qqnnttttttttfffdddddddccr`

J: Guest tertiary mirror computer address (usually an upper-case letter)

qq: Two-digit number of EDS messages left in the guest queue

nn: Two-digit number of characters in the message (in the underlined section). 00 if no message available.

tttttttt: Eight-digit message time stamp (no punctuation), with two-digit hour, two-digit minute, two-digit second, and two-digit hundredths of a second.

fff: Three-digit message number. Message numbers from 0-799 denote errors, 800-899 are numberical data formats, and 900-999 are successes.

ddddddd: Variable length message data section. For error and success messages, typically a simple text message. For numeric data formats, a combination of ASCII, decimal, and hexadecimal characters/digits, with the format being determined by the particular message number.

cc: Checksum, described above.

r: ASCII charater 13, a carriage return.

3: Repeat Last EDS Message

Commands the guest computer to re-send the last message it sent (implying that the host computer had a serial communication error during the last reponse). The guest's internal pointer should remain unchanged.

Command Format: `:J3r` (note that this command has no checksum)

Response Format: `~J3qqnnttttttttfffdddddddccr`

J: Guest computer address (usually an upper-case letter)

qq: Two-digit number of EDS messages left in the guest queue

nn: Two-digit number of characters in the message (in the underlined section). 00 if no message available (there is no time stamp, message number, or data in this case).

tttttttt: Eight-digit message time stamp (no punctuation), with two-digit hour, two-digit minute, two-digit second, and two-digit hundredths of a second.

fff: Three-digit message number. Message numbers from 0-799 denote errors, 800-899 are numberical data formats, and 900-999 are successes.

ddddddd: Variable length message data section. For error and success messages, typically a simple text message. For numeric data formats, a combination of ASCII, decimal, and hexadecimal characters/digits, with the format being determined by the particular message number.

cc: Checksum, described above.

r: ASCII charater 13, a carriage return.

4: Set UT

Commands the tertiary mirror computer to set its clock to the Universal Time given in this command. The control computers keep their clocks synchronized to GPS-provided universal time in this way.

Command Format: `:J4ttttttttccr`

Response Format: `~J4er` (note that this response has no checksum)

J: tertiary mirror computer address (usually an upper-case letter)

tttttttt: Eight-digit universal time (no punctuation), with two-digit hour, two-digit minute, two-digit second, and two-digit hundredths of a second.

cc: Checksum, described above.

r: ASCII charater 13, a carriage return.

e: Error flag: 0 if an error occurred, 1 if OK.

9: Free-form Command

Sends the tertiary mirror computer a free-form command, typically similar to the commands entered via the tertiary mirror computer's keyboard. This is used to command moves, homes, etc.

Command Format: `:J9nnddddddccr`

Response Format: `~J9ennmmmmccr`

J: tertiary mirror computer address (usually an upper-case letter)

nn: Two-digit number of characters in the message (in the underlined section). 00 if no message available.

ddddddd: Variable length free-form command section. This section will contain a command parseable by the guest computer, such as "MOVE 1000"

cc: Checksum, described above.

r: ASCII charater 13, a carriage return.

e: Error flag: 0 if OK, 1 or higher if an error occurred.

mmmm: Variable length diagnostic message (such as "Move ignored, brake on"). This should be printed out by the host computer in the command input box if the command was given by the user, or in the system log if the command was given by an automated routine in the host program. nn=00 in the response if there is no diagnostic message.

Log Message System

6

6 Log Message System

6.1 Errors Log Messages (0-799)

001: "SetStepperPhase: xxx bit not responding"

While enabling power to the listed stepper driver, the steppers Monitor bit was not responding to commanded steps. Check stepper driver wiring.

002: "Mirror motion timeout"

The mirror was moving up or down, but didn't hit a limit within a certain timeout period. A limit switch may not be responding.

003: "Can't move mirror up: ADC Turret in wrong position"

The ADC Turret must be in the "No ADC" position to leave room for the mirror to flip all the way up.

004: "M3 Turret: Lock pin timeout"

The M3 Turret lock pin failed to hit a limit in its timeout period. Check its limits and motion for proper operation.

005: "M3 Turret: Unable to unlock pin"

The M3 turret lock pin didn't unlock when commanded to. Try unlocking it manually (MT UNLOCK), and check limit switches, motion, etc.

006: "M3 Turret: User interrupted motion"

A keypress interrupted the turret's move.

007: "M3 Turret: Hit wrong limit"

During a move, the limit opposite the direction we should have been moving was triggered. Could be caused by limit line noise, or wiring errors.

008: "M3 Turret: Motion error"

This message shouldn't appear: it means a move routine returned an undefined error code. If this happens, there's a bug in the code.

009: "M3 Turret: Couldn't find index"

The program executed moves that should have gotten us centered on an index, but the index wasn't detected. Check motor motions, index, inputs.

010: "M3 Turret: Hit an unexpected limit"

During a move that shouldn't have triggered either limit, the limit was triggered. Could be line noise or wiring errors.

011: "M3 Turret: Unable to lock pin"

The M3 Turret lock pin didn't lock when it was commanded to. Check lock limits, lock motion, and turret position (make sure the turret was in a valid position for the lock to occur).

012: "M3 Turret motion ignored, Locked"

A move of the M3 Turret was attempted while the lock pin was in. Check the lock operation and status.

013: "M3 Turret motion ignored, already at limit"

A move of the M3 Turret was commanded towards a limit that was already set. Check turret position and limit inputs.

014: "M3 Turret motion halted, hit a limit"

A limit was hit during a move. Check turret position and limits.

015: "M3 Turret motion ignored, already on an index"

A command to move to and center on an index was given when the turret was already on an index. Try re-homing, or check index inputs.

016: "M3 Turret motion ignored, Home required"

A commmand was given to go to a new position when the system was lost. Perform a M3 Turret Home operation.

020: "ADC Turret: Lock pin timeout"

The ADC Turret lock pin failed to hit a limit in its timeout period. Check its limits and motion for proper operation.

021: "ADC Turret: Unable to home, mirror not down"

The M3 mirror must be in its fully down position for the ADC turret to home. This prevents damage to the mirror.

022: "ADC Turret: Unable to unlock pin"

The ADC turret lock pin didn't unlock when commanded to. Try unlocking it manually (AT UNLOCK), and check limit switches, motion, etc.

023: "ADC Turret: User interrupted motion"

A keypress interrupted the turret's move.

024: "ADC Turret: Hit wrong limit"

During a move, the limit opposite the direction we should have been moving was triggered. Could be caused by limit line noise, or wiring errors.

025: "ADC Turret: Motion error"

This message shouldn't appear: it means a move routine returned an undefined error code. If this happens, there's a bug in the program.

026: "ADC Turret: Hit an unexpected limit"

During a move that shouldn't have triggered either limit, the limit was triggered. Could be line noise or wiring errors.

027: "ADC Turret: Couldn't find index"

The program executed moves that should have gotten us centered on an index, but the index wasn't detected. Check motor motions, index inputs.

028: "ADC Turret: Unable to lock pin"

The ADC Turret lock pin didn't lock when it was commanded to. Check lock limits, lock motion, and turret position (make sure the turret was in a valid position for the lock to occur).

029: "ADC Turret: Bad index"

We landed on an index when we should have been at the forward limit. Check ADC turret motion, position and index detection.

030: "ADC Turret: Motion error"

The ADC turret should have moved to its forward ("No ADC") limit, but the limit never got activated. Check limit switches, turret motion and position.

031: "ADC Turret motion ignored, Locked"

The ADC turret was given a command to move while the lock pin was locked. Unlock with "AT UNLOCK"

032: "ADC Turret: Unable to move, mirror not down"

To prevent damage to the M3 mirror, the mirror must be in the down position before moving the ADC turret.

033: "ADC Turret motion ignored, already at limit"

A move of the ADC Turret was commanded towards a limit that was already set. Check turret position and limit inputs.

034: "ADC Turret motion ignored, already on an index"

A command to move to and center on an index was given when the turret was already on an index. Try re-homing, or check index inputs.

035: "ADC Turret motion ignored, Home required"

A commmand was given to go to a new position when the system was lost. Perform a ADC Turret Home operation.

036: "ADC Turret motion ignored, already at index"

A commmand was given to go to a new position out of index but there is mechanical limit. If the step value is less than 0, then this message will be sent. Try to move to position with steps greater than 0.

037: "ADC Turret motion ignored, locked"

A commmand was given to execute homing routine for a second time, but in the first time the “AT Home” command closed lockpin. Try to open lockpin (“AT unlock”), and try to execute homing procedure again.

040: "ADC Rotator: User interrupted motion"

A keypress interrupted the rotator's move.

041: "ADC Rotator: Couldn't find index"

The ADC tried to home to the index, but never found it. Check the index inputs and rotator motion.

042: "ADC Rotator motion ignored, Home required"

A command to move to a position was given, but the system had not been homed. Use the "ADCA HOME" or "ADCB HOME" commands.

043: "ADC Rotator Home ignored, move in progress"

ADC rotator movement is already in progress.

050: "DGH x Read Error"

There was a communication failure with the listed DGH module. Check communication with a terminal ("T2"), check DGH power, connections, etc.

051: "DGH x Error Messages Suspended"

There were more than a few errors, so further errors will not be displayed until a successful DGH read occurs.

052: "Tilt x: User interrupted motion"

During a tilt motion a key was pressed

053: "Error reading DGH x setup"

There was a communication error reading the given DGH's setup. Make sure the DGH has power (24V), is connected properly, and check communication with a terminal ("T2").

054: "DGH x setup wrong, expecting yyyyyyyy"

The listed DGH module has an incorrect setup string. Check its setup with a terminal ("T2"), and change it to match the given setup string.

070: "M3 Turret Pin: Unable to unlock, mirror not down"

071: "M3 Turret: Home Aborted, mirror not down"

074: "M3 Mirror: Flip down aborted, M3 turret pin not engaged"

M3 Mirror will not move if M3 or ADC turret pins are not locked.

076: "M3 Mirror: Flip up aborted, only allowed in AUX2 position"

M3 mirror movement isn't allowed, unless it is in AUX position.

500: "ADC Turret: Home aborted, other moves in progress"

505: "Tilt servo timeout"

6.2 EDS Log Message

800: MESA (LVDT) Analog reading update

Format: `800Cccc.cccBbbb.bbbAaaa.aaaTxxx.xxx`

ccc.ccc: data in volts for C DGH.

bbb.bbb: data in volts, for B DGH

ccc.ccc: data in volts, for A DGH

xxx.xxx: it is +12VDC

801: All digital I/O update

Format: `801DaabbccddeeffPxx`

aa, bb, cc, dd, ee, ff: hexadecimal field to log I/O digital data,

xx: hexadecimal field for parallel port values

802: DGH reading update (Load cell reading)

Format: `802ACaaa.aaaABbbb.bbbBCccc.ccc`

aaa.aaa : Data in volts for A-C load

bbb.bbb: Data in volts for A-B load

ccc.ccc : Data in volts for B-C load.

803: Subsystem position reading

Format: `803PabcdefAggg.gBhhh.h`

a: M3 Mirror

0: Mirror midway between limits

1: Mirror all the way down (in the beam)

2: Mirror folded up out of the beam

3: An error reading the limits

b : M3 Mirror Lock

0: Unknown

1: Locked

2: Unlocked

3: Error

c: M3 Turret

0: Unknown (lost or not homed)

1: Midway (in motion)

2: NAS E

3: AUX 1

4: AUX 2

5: AUX 3

6: NAS W

d: M3 Turret Pin

0: Midway

1: Locked

2: Unlocked

3: Error reading limits

e: ADC Turret

0: Unknown or not homed

1: Midway (in motion)

2: "No ADC"

3: ADC in position

f : ADC Turret Pin

0: Midway

1: Locked

2: Unlocked

3: Error reading limits

ggg.g: ADC A position (degrees)

hhh.h: ADC B position (degrees)

804: Target pressure and vacuum pulse width in ms

Format: 804VvvvvPpppp

vvvv: Vacuum pulse width value

pppp: Pressure pulse width value.

805: P,I,D Pressure/Vaccum control

Format: 805Pppp.pppIiii.iiiDddd.ddd

ppp.ppp: Proportional value

iii.iii: Integral value

ddd.ddd: Derivative term

6.2.1 Clay Telescope

800: Analog reading update

Format: 800C0Dccc.cccC1Dbbb.bbbC2Daaa.aaaC3xxx.xxx

ccc.ccc: data in volts for a DGH.

bbb.bbb: data in volts, for b DGH

aaa.aaa: data in volts, for c DGH

xxx.xxx: it is +12VDC

801: Digital I/O information

Format:801Daaab

aaa:

b:

802: Log DGH reading status

Format: 802DiRxxx.xxx

i: DGH A,B or C

xxx.xxx: DGH status value

803: Tilt motion information

Format: 803PT0aPT1bPT2c

a , b, c : The three tilt motion status

0: Stop

1: Out, pushing the mirror

2: In, pulling the mirror

804: Log servo parameters for m3 support

Format: `804Sabbb.bbbppp.pppiii.iiiddd.ddd`

a: Pressure/Vacuum control mode

0: Off

1: Auto

bbb.bbb : Pressure/Vacuum current average reading in mV

ppp.ppp : Proportional value

iii.iii : Integral value

ddd.ddd : Derivative term

805: Target pressure and vacuum pulse width in ms

Format: `805Wvvvvpppp`

vvvv : Vacuum pulse width value

pppp: Pressure pulse width value.

6.3 Success Log Message (900-999)

901: "ADC Turret: Home successful"

902: "ADC Turret: No ADC position successful"

903: "ADC Turret: ADC position successful"

971: "Servo first pass completed"

972: "Servo second pass completed"

981: "TiltA servo completed"

982: "TiltB servo completed"

983: "TiltC servo completed"

997: "M3.INI read success"

The M3.INI configuration file was read without problems.

998: "DGH x Error Messages Resumed"

The listed DGH module responded successfully after several previous errors.

999: "Program startup (vx.xx) mem=yyyk"

First log entry at program startup. Includes program version number and free memory left.

System Details

7

7 Tertiary Mirror Control System differences between Magellan I and II

Magellan I has two turret, one is for ADC motion and the other is for M3 motion. There is a computer inside M3 turret. This machine controls all M3 and ADC movements. This includes:

1. ADC turret
2. M3 turret
3. Tilts
4. Pressure/vacuum control
5. Up and Down mirror movements
6. ADC movements

Magellan II has its tertiary mirror inside cassegrian rotator. A local computer unit inside cassegrian, controls tilt movements and pressure/vacuum for M3. There is no ADC and M3 turret for Magellan II. The system doesn't need to control ADC / M3 turret nor ADC motion.

7.1 Commands

The following commands are not valid for Magellan II, M3 control program:

1. **ASP** , toggle power to the ADC rotators steppers

2. **MT** *<cmd>* and **AT** *<cmd>*, for turret movements

3. **AT** *<cmd>* and **AT** *n*, ADC turret motion

4. **ADCA/ADCB** *<cmd>*, ADC rotator motion

Troubleshooting

8

8 Troubleshooting