

Guide to PMAC and Associated Procedures

Table of Contents

Introduction	1
References.....	2
Low-Level Startup Procedure	2
Telescope Rack Description	2
Startup Procedure.....	3
PMAC Installation.....	4
PMAC setup	4
Telescope Specifics	4
Channel Mappings.....	4
Encoder Scaling Constants.....	5
Editing Variables and PEWin32 Control	6
Watch Window and GUI Specific Tools.....	7
PID Tuning through PMAC.....	7
Storing Changes.....	7
Code examples for different routines.....	8
Diagnostics	8
Homing of Azimuth and Elevation.....	8
Jogging	9
Feed Override Modification.....	9
High-Level Control.....	9
Appendix	10

Introduction

This document aims to consolidate a number of resources pertaining to the configuration and use of the PMAC and PEWin32 in a servo control environment. It contains information relevant to any installation housing a PMAC board but is written mainly to be specific to the 1.8m Icestorm telescope as will become apparent.

The procedures here are not extensive and point to the required external documents where relevant to avoid redundancy. Reading and following all procedures precisely will minimise mistakes and loss of time. Notes should be taken if changes to things such as PMAC RAM variables are made so that a reversion is possible.

Finally, the procedure does not cover a lot of the dome specific procedures or use of high level GUI clients such as VCP or the Operator Interface and instead focuses on the use of PMAC to control the telescope. Seek other documentation on these areas if required.

References

Much of the information in this document refers heavily to the following:

- **W:QA\Site Technical Procedures\General\Procedures\PMAC Software Installation Procedure.doc**
- **W:\QA\Site Technical Procedures\General\Procedures\PMAC Tuning Procedure.doc**
- **W:\QA\Site Technical Procedures\General\Procedures\Generic Telescope Control Procedures EOST.doc**
- **I:\Projects\Stromlo III\1.8m Telescope\1.8 Meter telescope and Enclosure Operation Procedure.doc**
- **PMAC User and Software Reference – Delta Tau**

Anything not fully explained here will (hopefully) be clarified in one of the above documents.

Low-Level Startup Procedure

Telescope Rack Description

The main control board is at the base of the rack at the rear. Once powered on, the internal PC powers up, along with some ancillary circuits. On the front panel, the large 3U panel with a switch mounted on it is the DIOP panel. This switch enables encoders and DIO devices on the telescope, allowing servo control.

If the DIOP is powered, the three right hand lights should come on, being *PS_1*, *PS_2* and +5V Power. Other lamps may also light if ESTOPs are active, if any axis is in its outer limit position, or if the Failsafe Override key is held on. The Failsafe Override key is used in outer limits to carefully drive the axis into its operating range again. This should not often happen.

If any ESTOP condition cannot be cleared, it is risky but possible to disconnect the ESTOP cable from the CJ5 connector at the rear, and bridge out pins 2 and 6. This is a dangerous procedure, not to be performed without complete knowledge of the consequences. **SAFETY WARNING: EQUIPMENT.**

A lamp test button is also provided on the DIOP panel.

The PC power switch is inside the hinged cover of the PC front panel. WIN NT 4 will boot when power is applied. A keyboard must be connected.

Anders Ridley-Smith
December 2011

GPS signals are also needed for correct operation. The signals used are IRIG-B, 10 MHz and 1 PPS.

The top two panels contain the drive amplifiers, which feed the motors. Encoder Planar 0 is at the top, and contains the secondary drive amplifiers. These are, from the left, *M2C*, *M2B* and *M2A*. Below this is Encoder Planar 1, containing from the left, *EI*, *Az B* and *Az A*.

The second LED from the top is lit when the amplifier is powered whilst the top green LED indicates that it is enabled, ready to control the motor current. Note that Azimuth drives A and B are driven in parallel.

Below the PC and DIOP panels is the PI Controller panel, which drives the fast tip/ tilt mechanism on the secondary drive. Finally, at the bottom of the rack is the Daytronic temperature sensor controller module.

Startup Procedure

1. Check the dome drive ring for any debris on the support roller drive ring.
2. Check levels 1 and 2, and ensure that there are no obstructions to rotation, such as ladders leaning on the pier, extension cables into the telescope base. Or a GPS temporary cable outside the enclosure from level 1.
3. Check that the telescope is free to move in azimuth, specifically:
 - a. The red tow hitch bracket has been removed.
 - b. The mirror cart rails from Level 3 floor to the telescope yoke are removed.
4. Push the telescope to the centre of its range, such that the tow hitch is central between its shock absorbers.
5. Connect all cables from the cabinet to the base panel, and have them checked.
6. Switch on the cabinet at the rear circuit breaker. The PC should begin to boot.
7. Pull out all Emergency Stop (ESTOP) buttons.
 - a. The light in a button flashes when it is active, i.e. pushed in.
 - b. If the light is ON steadily, then a different button is pushed in.
 - c. If the light is off, then all ESTOP buttons are disengaged.
8. Switch on the DIOP switch on the front panel to enable the encoders. The three right most lights on the respective panel should light.

PMAC Installation

If it is required to do a fresh install of either the PEWin32 application or the ALLFILES.PMC software onto the PMAC, a detailed description of the required procedure can be found at **W:QA/Site Technical Procedures/General/Procedures/PMAC Software Installation Procedure.doc** (or similar).

PMAC setup

Once it is confirmed that a valid version of the PEWin32.exe application is installed on the PC, as well as the most recent version of ALLFILES.PMC, the PMAC is ready to be initialised for control.

1. Log into the PC and find the PEWin32 application (the shortcut should be on the desktop).
2. With a watcher at the telescope, near an ESTOP button, run PEWin32.
3. Disable all default startup PLC programs with "Ctrl+D".
4. Startup all required PLCs by typing "enable PLC01" in the command line of the PMAC Executive. Note that PLC01 is a bootstrap program that enables all the subsequent PLCs. It also initialises all the required variables and as such can be thought of like hitting a reset button.
5. Although not necessarily PMAC related, any movement of axes needed in the following step requires that the azimuth of the **dome** (note: different to telescope azimuth) be enabled. This is done by selecting the 'Azimuth Enable" option in the higher level Operator Interface.

Telescope Specifics

The following data is specific to the Icestorm 1.8m Telescope Enclosure. If other telescopes are being configured, refer to the more complete data provided by the Generic Telescope Control document found at **W:QA/Site Technical Procedures/General/Procedures/Generic Telescope Control Procedures EOST.doc**

Channel Mappings

Inside the PC is the PMAC motor controller circuit. The Turbo PMAC I has 16 axes of output control, and 16 encoder input channels. These channels are matched as follows:

Channel Number	Output Channel Usage	Encoder Channel Usage
1	Secondary drive A	Secondary drive A

2	Secondary drive B	Secondary drive B
3	Secondary drive C	Secondary drive C
4		
5		
6		
7		
8	-	1 MHz derived master time signal*
9	Azimuth drive (A & B)	Az Read Head A
10		Az Read Head B
11		Az Read Head C
12		Az Read Head D
13	Elevation drive	EI Read Head A
14		EI Read Head B
15		EI Read Head C
16		EI Read Head D

Figure 1: Keck OT 1.8m Telescope - Use of Available PMAC Channels

Note that Channel 8 has no drive output, but the encoder Channel 8 is used to receive a 1 MHz timing clock, the reference clock for all real time operations. The DIOP circuits contain a divider circuit for reducing the 10MHz GPS clock to 1 MHz for this purpose.

In the Keck OT implementation, another signal is often used, for applying camera End of Integration signals to the PMAC card. The purpose of this signal is to make the PMAC latch its encoders and time channel for use by other PCs when camera operations are required to be synchronised to current telescope pointing data. This signal is accurate to the internal loop speed of the fastest PLC program, PLC 0. This speed is measured at 2.2 kHz, or 440microseconds.

Encoder Scaling Constants

The encoder drums and tapes have a specific number of cycles built into them. Each drum has a designed tape length and number of cycles associated with it. This number of cycles per drum revolution correlates with the number of encoder counts per encoder per degree revolution. Based on the number of encoders per axis, the number of encoder counts per degree adds by each encoder.

For the Keck telescopes, the azimuth and elevation encoder drums have the same diameter, and the same number of read heads per axis. Therefore the same number of encoder counts per degree apply to each axis. It is important to be aware of the scale to understand what a change in encoder counts roughly means in terms of degrees. This will allow the user to interpret how fast the telescope is moving around an axis or its current position.

	Azimuth Scaling Constant (counts/deg)	Elevation Scaling Constant (counts/deg)
Keck OT	2364444.4444	2364444.4444

Stromlo III 1m	1671111.1111	1457777.7777
Beijing 1m	835555.5555	728888.8888

Figure 4: Tape based Scaling Constants

For more specific data including the Secondary Axis scales, refer to the Generic Telescope Control document as listed above.

Editing Variables and PEWin32 Control

Manual mode operation (operating without the PLCs) is not recommended for this telescope, despite being used on the old Keystone system. Instead, PLC control is recommended at all times. Use of “Ctrl+d” to disable the PLC operations is not recommended as certain protections are no longer in place. If PLCs have been disabled, type “enable PLC01” to restart all PLCs using the PLC01 bootstrap (master) PLC program.

Explanations of I variables as well as key command inputs can be found in the PMAC User and Software references, found in the SLR station. For definitions of certain additional variables refer to the Appendix of this document. Further to that, observing the text of `allfiles.pmc` will provide a great deal of insight into what specific variables are used for as well as how individual PLCs function.

M variable, P variables and I variables are still used inside the PLC programs which control the telescope. These variables are:

- **M variables** - memory indexing variables to the dual port RAM
- **P variables** - private variables, with no conventions of use, for user custom activities
- **I variables** - mapped memory variables with standard conventions for numbering, which affect PMAC operation in specific ways (e.g. Tuning parameters)

With PEWin32 running, commands entered can alter the values of M, P and I variables in RAM. Rather than providing superfluous and random examples here, refer to the later section for examples of how to edit specific variables and carry out specific tasks.

Editing M, P and I variables can activate certain functions and PLCs or change the operating parameters of the system. As such, logging what has been entered in the PMAC Executive is a good idea.

Further to altering variables, servo control can be done through the command line by using certain PMAC-specific commands such as jogging with feedback control or driving servos with constant torque/current. More details of these commands are given in the Appendix and obviously are more fully explained in the PMAC Software and User Reference. These are physical format manuals that are available from the SLR Station. See Chris Moore.

Watch Window and GUI Specific Tools

Whilst editing variables or carrying out procedures with the PMAC Executive, it may be useful to display certain variables on the Watch window of PEWin32. Adding a variable can be done by selecting the window and hitting insert, followed by typing in the required variable name/number.

Jogging commands can be entered manually, as explained in the Appendix, but they can also be sent using the sub-GUI found at View>Jog Ribbon. This tool can be used to send the servos to a certain count/position or to jog indefinitely at a certain rate and so on. The key FYI here is the use of the bottom drag-bar that alters the variable entitled "FEED OVR" or the feed override.

It has been found that sometimes this has been at 0%, resulting in non-responsive jog commands. If this is at 0%, jog commands will not work. Dragging this to 100% will enable jogging with the set drive speed (Set by variables I922, I1322 etc). A setting of, say, 50% appears to cause jogging to be at a rate of 50% of the drive speed parameter. Once the bar is set, the ribbon may be closed and all jogging (and homing) functionality should be returned.

Alternatively, the feed override can be altered on the command line by using the "%" command. A feed override value of 100% is usually desired as it provides real-time execution of jogging commands as normal. Details of this in the code example section.

PID Tuning through PMAC

If it is required that the PID control of certain axes be altered due to poor performance, reference should be made to the PMAC Tuning Guide found at **W:QA\Site Technical Procedures\General\Procedures\PMAC Tuning Procedure.doc**. This document outlines much of what is contained in this document as well as more specific procedures for tuning. In addition, it contains a step by step procedure for using the Detailed Plotting tool in PEWin32.

Storing Changes

Any changes made by altering a variable in the PMAC Executive, such as setting a new value for I922, will only be made to the RAM value of that variable. As such it is volatile and will not be stored upon a system reset. Further, any reset of the PLCs such as by typing "enable PLC01" will restore default values for all variables as specified in the `allfiles.pmc` code file.

To cause a permanent change to a variable, the `allfiles.pmc` code must be altered directly and the file reloaded onto the PMAC's ROM.

This means that in the case of an operation such as PID tuning, it is a good idea to first alter the variables in RAM by changing them in the command line and then testing if these changes produced a desired result. If so, the variables can be identified in `allfiles.pmc` and changed permanently. Instructions on how to then download this new version of code (and protocols to follow) can be found in the installation procedure document at **W:QA/Site Technical Procedures/General/Procedures/PMAC Software Installation Procedure.doc**.

Code examples for Specific Routines

The following section contains various code examples for carrying out some of the most frequently required routines. They should not be copied verbatim unless the user fully understands each line and is aware of how to check the appropriate variables. With certain error flags set the following examples may not work. Note that the ">>" character indicates output from PMAC to the display.

Diagnostics

```
#           ; What is the current axis?
>>13       ; Current axis is elevation (Ch 13)
M3600      ; Check Azimuth Amplifier
>>0        ; Azimuth amplifier not enabled
p          ; get current position
>>27349302 ; Elevation is currently at said position
           (counts)
#9         ; Make Ch 9 (Azimuth) the current axis.
p          ; get current position
>>-12426443 ; Azimuth is currently at displayed
           position (counts)
I922       ; What is current drive rate (counts/msec)
           of azimuth
>>4728     ; Drive rate is 4728 counts/msec
M3625      ; Is azimuth homed?
>>0        ; No, it is not.
M3630      ; Is there an error in the Azimuth axis?
>>0        ; No
M3680      ; Is there an error in the Elevation axis?
>>0        ; No
```

Homing of Azimuth and Elevation

```
M3600=1    ; Enable Azimuth amplifier
           (Enable dome azimuth)
M3601=1    ; Begin azimuth homing (wait for
           completion)
```


Anders Ridley-Smith
December 2011

```
M3625          ; Azimuth Homed?  
>>1          ; Yes  
M3650=1       ; Enable Elevation amplifier  
M3651=1       ; Begin elevation homing (wait for  
              completion)  
M3675          ; Elevation Homed?  
>>1          ; Yes
```

Jogging

```
M3600=1 M3650=1; Enable Azimuth and Elevation Amplifiers  
#9           ; Select Azimuth axis  
I922=2364   ; Set jog speed to 1 deg/s (Recall scaling  
            constants)  
J+          ; Jog in positive direction at speed set  
            above  
j/         ; Cease jogging  
j=20000000 ; Jog to set count position  
j/         ; Cease jogging  
#13        ; Select Elevation Axis  
I1322=2364*2 ; Set jog speed to 2 deg/s  
j:30000    ; Jog 30000 positive from where axis is  
            now  
j+         ;  
k          ; Kill current axis (Elevation)  
CTRL+k    ; Kill all axes  
M3600=0 M3650=0; Disable Axes
```

Feed Override Modification

```
%           ; What is the feed override value?  
>>20       ; Feed override is equal to 20 percent  
%100       ; Set the feed override to 100%  
%          ; Check feed override  
>>100      ; As expected  
H          ; Hold feed override (equivalent to "%0")  
%  
>>0
```

Refer to the Appendix for a summary of commands.

High-Level Control

The high level control programs such as those installed on the PC in the SLR station, including VCP and the Operator Interface (OI), operate by modifying variables and flags and activating certain PLCs just as done above (e.g. PLC04 for Jogging, PLC11 for homing etc). It is important to keep in mind that when operating at the lower level through the PMAC Executive, any flags set and errors that occur will be reflected in the status of the VCP and OI. It is a

good idea to ensure that the system is returned to a normal state before high-level control is returned. If various flags are set that have arisen from mistakes made in the PMAC Executive (These should be logged in a notebook and attempts to reverse them should first be made) a reset of the PLCs is often wise (“enable PLC01”).

Furthermore, note that the VCP operates through an intermediary file, **telescope.ini**, which sets various parameters. This means that carrying out slew movements in the VCP is very different to carrying out a jog movement on the PMAC Executive *even if the start and finishing angles are the same*. The reason for this is that the slew movement command from the VCP initiates a trapezoidal movement that differs in implementation from the simpler jog command. Keep this in mind when testing for the system performance such as reviewing changes made to PID tuning parameters.

Appendix

The following tables list useful commands for the control of the telescope using the PMAC Executive as well as the most commonly altered variables and most commonly checked/modified flags.

Command	Function	Comment
\	Hold program	Slows all drives to stop via main clock stop but permits jogging
%	Report Current Feed Override Value	
%<NUM>	Modify Feed Override	e.g. %100<Enter> sets feed override to 100% (real-time operation)
H	Hold feed	Sets feed override value to 0 so no jog commands execute.
a	Abort present program	
k	Kill current axis	
Ctrl+k	Kill all axes	Stop motion suddenly
#	Display selected axis	E.g. #<Enter> to get selected axis
#<NUM>	Change selected axis	e.g. #9<Enter> to select azimuth axis
Mxxxx=1	Set Memory flag xxxx	Flags used for enabling drives and other binary operations, e.g. M3600=1 enables azimuth drive axis via amplifier enable.
Ixxxx=yy	Edit I variable value	Values used for servo parameters, e.g. I930=40000 sets proportional gain in azimuth drive to 40000
o15	15% open loop output	Applies to currently selected axis only
J15000	Jog 15000 counts positive direction	Currently selected, homed and enabled axis
J=-8000000	Jog to encoder count value -8000000	Currently selected, homed and enabled axis
j/	Stop jogging	

J+	Jog in positive direction	
Ctrl+d	Disable all PLCs	Will cause all PLC control to be disabled. Not advised.
Enable PLC01	Enable bootstrap PLC	PLC01 is a bootstrap that activates all subsequent PLCs in <code>allfiles.pmc</code>

Useful PMAC Executive Commands

Variable	Operation	Comments
M3600	Enable/Disable Azimuth Amplifier	e.g. M3600=1 to enable
M3601	Home Activation Flag	e.g. Set M3601=1 to begin homing sequence in azimuth axis
M3625	Homing Complete Flag	Check value of M3625 to determine if homed (=1)
I922	Azimuth Jog speed (counts/msec)	e.g. I922=1000
M3602	Homing Abort Flag	e.g. Set M3602=1 to abort homing sequence
M3607	Error Acknowledged Flag	e.g. Set M3607=1 to acknowledge errors and clear error flags
M3630	Axis Error flag	If M3630=1, an error has occurred (in some way) for that axis
I930	Proportional Gain	
I931	Derivative Gain	
I932	Velocity Feed Forward Gain	
I933	Integral Gain	

Commonly Used Variables

It should first be said that whilst the above variables are of type M, many are mapped to a P variable with the same number. Sometimes, editing must be done through the P variable. A review of the source file will help to determine this (Ctrl+F is your friend).

Note that the above table is specified for the azimuth axis. I Variables are defined such that the first to numbers following the I relate to the channel number of the drive. For example, I1322 is the jogging speed for the elevation axis. M variables in `allfiles.pmc` are defined such that each drive (Azimuth, Elevation, Secondary A, Secondary B and Secondary C) are offset by a value of 50. As such, by adding 50 to the M value of any variable in the above table you will get the variable for the next drive. For example, M3650 is the Enable/Disable flag for Elevation. For reference, the Enable/Disable variables for each drive are separated as follows:

Drive	Enable/Disable Variable
Azimuth	M3600
Elevation	M3650
Secondary A	M3700
Secondary B	M3750
Secondary C	M3800

Drive Variable Offsets

It is advised that for more understanding the user has a detailed look at the `allfiles.pmc` code.