

The Generic Pixel Server Dictionary

Nick C Buchholz*, Philip N. Daly
National Optical Astronomy Observatory, Major Instrumentation Group,
950 N Cherry Ave., Tucson, AZ 85719-4933;

ABSTRACT

Instruments and telescopes being planned for the US community include a wide assortment of facilities. These will require a consistent interface. Existing controllers use a variety of interfaces that will make using multiple controller types difficult. A new architecture that takes maximum advantage of code and hardware re-use, maintainability and extensibility is being developed at NOAO. The MONSOON Image acquisition/Detector controller system makes maximum use of COTS hardware and Open-Source development and can support OUV and IR detectors, singly or in very large mosaics. A basic requirement of the project was the ability to seamlessly handle even massive focal planes like LSST and ODI.

Software plays a vital role in the flexibility of the MONSOON system. The author's have built on their experience with previous systems (E.g. GNAAC, wildfire, ALICE, SDSU etc.), to develop a command interface, based on a dictionary of commands that can be applied to any detector controller project. The Generic Pixel Server, or **GPX**, concept consists of a dictionary that not only supports the needs of projects that use MONSOON controllers, but the set of commands can be used as the interface to any detector controller with only modest additional effort. This generic command set (the **GPX** dictionary) is defined here as introduction to the **GPX** concept.

1. INTRODUCTION

In the past 18 years the authors have written, modified or used more than 12 separate and distinct hardware/software systems designed to control detector focal planes and capture the data generated by them. The needs of the US community for new instruments and telescopes have highlighted the need for a hardware/software system capable of controlling a wide variety of focal planes. Proposals for instrument focal planes containing single detectors up to mosaic focal planes of 500 or more detectors have been developed. The limitations of current controllers have forced NOAO to start a new controller development project. This controller, called MONSOON, was from its inception designed to allow the control of any astronomical focal plane, IR FPA's, CCD's, Orthogonal Transfer Arrays (OTA's), etc. It was designed to be extensible so that the sets of electronics could be replicated to control massive homogeneous focal planes. A key feature of the MONSOON system design was that a single software system could be used without extensive modification to handle the various focal plane arrangements and detector types.

The Generic Pixel Server, or **GPX**, concept on which the MONSOON software is based was first discussed at the Nov. 2001 ACCORD conference in Santa Cruz. The concept introduced an interface, that described the interactions between an image acquisition system or pixel server and external systems and that could be reused for all of the new instruments, focal planes and/or detector types under consideration for development.

A proposal for the Generic Pixel Server interface definition was presented at the February 2002 AURA Software Conference held at the STSI in Baltimore, MD. Further development of that interface as NOAO Interface Control Document 4.0¹ was done in the spring of 2002 and a version of the complete **GPX** interface, including a software library which implements a simplified version of the interface commands, has been developed in conjunction with the MONSOON project software effort. With very few exceptions the original interface definition presented in 2002 has held up to the rigors of actually writing code to control detectors and acquire data from IR and OUV focal planes.

In this paper we will present the current **GPX** dictionary, i.e. the set of commands that we have found sufficient to control the focal planes being developed at NOAO. Currently in use or under development are: engineering LAB systems for the development and characterization of individual IR and CCD detectors, the NEWFIRM instrument system controlling a 2 by 2 mosaic of 2048x2048 ORION_II detectors, a system for development and testing of the 4k by 4k OTA devices being developed for QUOTA and ODI, and a system for a Multi-Object fiber spectrometer.

The early stages of the MONSOON project included extensive consultation with the broader community to insure that the design decisions we made for MONSOON did not conflict with the needs of that community. In addition, NOAO has begun working with outside organizations on the use of the **GPX** concept in other instruments. At least three

* Further information email: NCB: nbuchholz@noao.edu, PND: pnd@noao.edu

science clients are being worked on by groups outside of the MONSOON project: a NEWFIRM Client being developed under DRAMA, an ArcView client being developed at CTIO and a client for the UIUC/Fermi Lab group's instrument.

2. Observatory System Model

The Generic Pixel Server (**GPX**) is a layer in an Observatory Control System Reference model, (GPX is in light gray), that corresponds to what has been called the detector controller and data capture/pre-processor. These systems deal with the details of controlling the focal plane detectors and delivering pixel and meta-data to the Observer. Many examples of observatory system models exist. The **GPX** can be considered a subsystem of the Instrument Control system, as in Figure 1, or it may be a sub-system of the Observatory control system directly. In either case the interface to the **GPX** can clearly use the same command set or dictionary.

The **GPX** interface provides for access and control of the data generation and capture hardware. It combines the functionality inherent in the detector head electronics (detector controller) and data pre-processor node. It presents a common interface to the upper level observatory systems regardless of the underlying detector technologies or focal plane arrangement being used for observing. Note that the Generic Pixel Server does not include provisions for the control of the telescope or the instrument mechanisms. Other layers of the observatory system handle those jobs. The **GPX** handles taking and controlling exposures, array temperature control if needed and, if one is provided, control of a local shutter.

OBSERVATORY SYSTEM REFERENCE MODEL

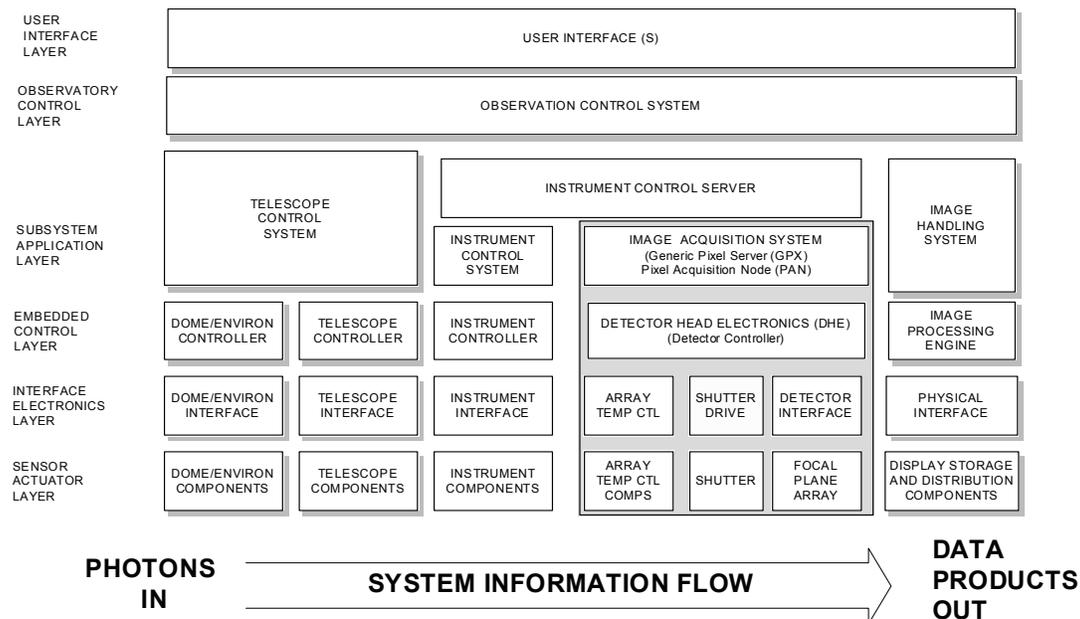


Figure 1. Observatory System Reference Model

3. GPX Philosophy

The original concept for the GPX developed out of discussions at the 2001 ACCORD conference that brought together a number of scientists, electrical and software engineers to discuss detectors and detector control and how to deal with the challenges being presented by the rapid development and deployment of new detector technologies. One of the conclusions of the final round-table discussion was that while different detector technologies might require different hardware for control, the set of commands to control all detectors types was essentially the same. Since NOAO was just beginning the design phase of the MONSOON project, we had an opportunity to test this conclusion.

A review of existing control systems was undertaken to determine the set of commands needed to control a detector focal plane for astronomy. This process was complicated by the fact that with the exception of the GNAAC control system designed and built for GEMINI, many earlier systems (wildfire, ArCon, ALICE and various SDSU systems)

were combined systems, that not only controlled the focal plane but also controlled the instrument mechanisms and telescope control program in the same software package. This meant a number of legacy commands would not be needed since MONSOON as proposed, was to be a pixel server, with no facilities for controlling filter wheels or telescope mechanisms. The GPX concept, as a generic server that acts like a commercial digital camera with a set of simple modes to control the system and a “take picture” shutter trigger, matched MONSOON exactly. This meant that the MONSOON system could be a GPX and we would only need to provide commands to control a focal plane and produce pixel data.

The philosophy of the GPX command set was to provide a minimal command set needed to conveniently control, configure and extract pixel data from a detector focal plane. From the observer’s viewpoint, the GPX is controlled by setting the mode of operation and controlling an exposure to produce useful observations. The GPX commands are designed to work together to give science data, and the commands are flexible enough to be used by all future instruments and focal planes.

3.1. Named Configurations/Modes

Part of the philosophy for the GPX was that the astronomer/user should not need to know the details of the detector/focal plane set-up in order to get “good” science from a system. In many cases the information needed to correctly set-up and configure the massive focal planes the GPX is designed to handle will be more than the astronomer/user can be expected to know. Imagine having to remember and set the 6-10 adjustable voltages available to get the best performance out of the 500 individual detectors in the LSST focal plane. For this reason the GPX assumes that the knowledge needed to configure the system is contained in the named modes. These modes conceal the details of the underlying detector or focal plane from the user.

The instrument scientist determines the modes needed to do the science. The detector engineer will then derive, through a detector optimization procedure, the precise voltage settings needed for each detector in the focal plane. These settings and the operating methods, timing waveforms and sequencing will then be used to achieve the science goals of the mode. The mode can set commonly known attributes like binning, pxlRows or intTime, or it may set names familiar only to the detector engineer like ACQ0_DAUGHTER or MCBSEQLOOPREGS [0], as in Example 1 below. The modes developed by the instrument scientist and detector engineer would define the science operations possible for the focal plane.

```
#Generic_I Initial Setup Mode File - Author – NCB – Date 20040123
# Attribute Name=Value                               #Comments
# Load array dependent Clock&Bias Board voltages generic_I sets up like an alladin_III detector
vnrw=-6.00                                           # Set VnRow to -6.0 volts
vncol=-4.38
vddout=-1.00
vset=-1.60

#COUNTER INITIALIZATION
MCBSEQLOOPREGS[0]=128                               # Load Row Loop Counter register
MCBSEQLOOPREGS[1]=32                               # Load Column Loop Counter register

# SETUP IR ACQUISITION Board Registers
ACQ0_DAUGHTER=1                                     # Enable the pipeline write function with Daughter board inactive
ACQ0_CFGADC=0                                       # Set the adc mode from bank 0

# these determine the array size must be smaller than the values in the .arr file
mosaicRows=1                                         # number of arrays in the col direction
mosaicCols=1                                         # number of arrays in the row direction
pxlRows=65535                                       # number of rows in one array
pxlCols=36                                           # number of cols in one array
```

Example 1 Simple GPX Mode Record

A named mode may contain all of the setup parameters and values needed to properly control the focal plane to produce “good” science data or it may contain only a subset of the values. E.G. The NEWFIRM instrument may provide a named mode called “LowNoise_LowBackground” which would set-up to take images of fields at wavelengths where the sky background was low and the observer knew that the highest possible signal to noise ratio was needed to do the science. Another mode “Bright_Point_Source” might be used which modified some of the parameters of the “LowNoise_LowBackground” mode to allow imaging on a field that included a very bright point source.

The named mode concept is also extended to include modes that are developed by the users to meet the needs of specific observing programs. In the previous example an attribute of the mode might be the number of digital averages to do on each pixel. The observer could set up a personal mode that set this value to 8 instead of the default 4. This new mode might be named “My_LN_LB_DA8” and the observer could then use this name to set-up the system. The addition of new modes of operation for new systems or observing techniques is easily done and because the modes are extensible the GPX command set should be usable into the future.

3.2. Common Command Structure

All of the **GPX** commands share a common structure. A **GPX** command consists of an ASCII string of size determined by the implementation, (MONSOON allows up to 4095 Bytes) and has several parts. The structure of the commands includes an optional eight byte command ID, the command name, zero or more optional directives, zero or more optional positional parameters, and finally zero or more optional attribute-value pairs.

```
[cmdidxxx] gpxCommandName [<DIRECTIVE>]* [Positional Parameter]* [Attribute-Value Pair]*
```

The current MONSOON implementation does not support the command ID field but this may be added a later date

3.3. Common Response

In addition to having a common command structure the GPX has a common response structure. A GPX response consists of an ASCII string the size of which depends on the specific GPX implementation, (MONSOON allows up to 4095 Bytes). The structure of the response strings is also common to all commands. The responses consist of an optional eight-byte command ID that matches the command ID of the command that initiated the response. This is always followed by a text error status response, followed by a simulation warning if the GPX is operating in simulation mode, followed by a string giving a human readable status message. Each GPX command returns a single command response within 100 ms of command initiation. This response may contain multiple newlines and multiple OK indications. The response has the format

```
[cmdidxxx] OK [[SIM]]: “String status response” or
```

```
[cmdidxxx] ERROR [[SIM]]: “String reason for error”
```

In addition certain commands, gpxStartExp for one, may generate a gpxAsyncMsg command at some later time. These asynchronous messages can announce a reboot, power glitch, a change in detector status requiring action by the upper level system, or the completion of some extended time command.

3.4. GPX command directives

All **GPX** commands can accept directives that modify the behavior of the command. While not all commands take action on all directives no command should fail because of an ignored directive. The list below is a partial list of the directives currently in use and which commands respond to them

- <SAVE> - accepted by all mode setting commands gpxSetMode, gpxSetIDPCfg, gpxSetExpCfg etc., this directive causes the system to read and save all of the attributes normally set by this mode command.
- <RESP> - used by the gpxAsyncMsg command/response to indicate the message requires a response from the upper level system.
- <INFO>, <EVENT> - used by the gpxAsyncMsg command/response to indicate the message needs no response and is either for information purposes or is a report of an event in the system.
- <CREATE> - accepted by all commands to allow the creation of new attributes (not yet implemented.)
- <REMOVE> - accepted by all commands to allow the deletion of attributes. (Not yet implemented.)
- <OVERRIDE> - accepted by all commands to allow the attributes in the DHE to be set in violation of standard restrictions.
- <CHKRANGE> - accepted by gpxSetAVP and gpxGetAVP to allow the client to determine if a value is in range for this attribute.
- <CHKACCESS> - accepted by gpxSetAVP and gpxGetAVP to allow the client to determine if setting or reading the attribute is permitted at the present time.
- <GPX> - accepted by gpxHelp to restrict the Help information search to only the **GPX** commands.

- <ATT> - accepted by gpxHelp to restrict the Help information search to only the attribute names.
- <NAMES> - accepted by gpxHelp to restrict output to command and attribute names
- <GUI> - accepted by gpxDump to retrieve information about the list of attribute categories the system integrator created within the system configuration record.
- <IMMD> - a directive that modifies the behavior of commands to make them immediate, i.e. executed at the time the command is issued. This applies to certain commands that have state or occur during the execution of another command like gpxShutter.

4. Command Classes

GPX commands fall into several categories that divide them according to function. In the sections that follow we discuss the various categories and outline the minimum GPX command set in the category. Every GPX conforming system will provide the commands in the minimum set. Additional commands may be added for the convenience of the system users and developers but complete functionality should be available using only the minimum command set. The MONSOON GPX interface includes several convenience commands that were added to the minimum set for the engineers doing hardware testing on the prototype MONSOON hardware.

4.1. System Set-up Commands

The first command category is the set of system setup commands. The minimum set in this category consists of three commands: gpxSetMode, gpxSetMemCfg and gpxSetAVPair. Note that each system is provided with a set of default configuration record names so that even gpxSetMode with no other information should produce a safe and valid setup of the system.

4.1.1. gpxSetMode

The gpxSetMode command is the main set-up command for GPX systems. The format of the command is:

gpxSetMode [<SAVE>] [*cfgFileName*] [arrConfig=arrFileName] [expConfig=expFileName] [idpConfig=idpFileName]

If the <SAVE> directive is present that command reads every attribute in the system and saves its current value in a configuration record with a name which is "*cfgFileName*".mod. This name is then the name of a new major mode for the current MONSOON system. Without the <SAVE> directive the command reads the configuration record *cfgFileName* and proceeds to set every attribute in the record to the requested value.

The command returns an OK or ERROR response indicating where any error took place.

The optional arrConfig, expConfig, idpConfig attributes are used to indicate which modes are to be used for the setting of the detector, exposure and image data preprocessing configurations.

4.1.2. gpxSetMemCfg

The gpxSetMemCfg command is provided to allow the rapid download of large chunks of DHE memory. In MONSOON this command is used to load the sequencer code and pattern memories in the system. The format of the command is:

gpxSetMemCfg [<SAVE>] [*memCfgFileName*] [*saveAddrRange*]

The format of the *memCfgFile* contains comment lines, address lines and memory block value lines. The result of the download of the data is that the memory addresses specified in the file are loaded into the DHE memory. The response is an indication of success or failure with an indication of the reason for the failure. If the <SAVE> directive is present the DHE memory in the address range will be saved to the memory configuration file named. (NOT yet implemented).

While it is possible to set up the entire DHE by using only this command, the disadvantage that the settings for DHE attributes would not be checked, stored or verified by the PAN level protection software makes this less than useful.

4.1.3. gpxSetAVP

The gpxSetAVP command allows a user to set an arbitrary individual parameter in the GPX. Any low level internal parameters available in the Generic Pixel Server should be settable with this command. This command will normally be used after the gpxSetMode has been executed. It will make changes to attributes like integration time or binning that the

observer may want to vary during a set of observations. A list of all the attributes this command can set should be provided with every GPX system developed.

4.1.4. Additional set-up and mode commands

The three commands `gpxSetArrCfg`, `gpxSetExpCfg` and `gpxSetIdpCfg` are provided in the MONSOON GPX to allow modes to be modified in smaller pieces. The attributes settable from these commands are a subset of the total attribute set and will allow the same base mode to be used with a variety of exposure, detector or data pre-processing modes. The attributes settable by each of these commands are determined by the Instrument Scientist and Detector Engineer at system integration.

4.2. Exposure sequence control commands

The next class of commands is the Exposure Sequence Class. The minimum set of these commands is: `gpxStartExp`, `gpxPauseExp`, `gpxResumeExp`, `gpxStopExp`, `gpxAbortExp`, `gpxWaitForTrigger`. These command control the sequence of events while taking data in a GPX. All of the minimum set should be provided but many systems may treat the `gpxPauseExp`, `gpxResumeExp` as NO-OPS. Systems without internal shutters, IR systems in general and others may choose to ignore the pause/resume commands. Also, while a system may execute a pause and later resume there is no intelligence built into the GPX to determine if the result of a sequence of pause/resume commands will result in valid science data, this is left to the observer or instrument scientist.

In the MONSOON implementation the detector library level pause/resume commands are used to ignore or execute a pause based on the focal plane the system is attached to.

4.2.1. `gpxStartExp`

This command sends the DHE the command to start an exposure. It is assumed that all required preconditions are set and all set-up routines have been run prior to issuing this command. Inherent in the low level functionality of the command are the collection of any time stamps or house-keeping data required and/or any shutter or voltage controls need prior to each exposure. The result of this should be of the form "OK exposure started" or "ERROR: startExp failed for reason A, B, C". When the exposure is complete, a `gpxAsyncMsg` will be sent to the upper level system to announce the completion of the exposure. During the interval between issuing the `gpxStartExp` and receipt of the `gpxAsyncMsg` announcing its end the client may issue a number of `gpxPauseExp`, `gpxResumeExp` commands and a single `gpxStopExp` or `gpxAbortExp`

4.2.2. `gpxPauseExp`, `gpxResumeExp`

These two commands allow a running exposure to be paused and resumed during its lifetime. Note that while the system makes every effort to insure that the final data are valid, there are no protections built-in to the system that will prevent a knowledgeable user from changing the state of the system in such a way as to create invalid data. These protections are left to the instrument scientists to institute.

4.2.3. `gpxStopExp`, `gpxAbortExp`

These commands stop the currently running exposure. In the case of the `gpxStopExp` command the system captures a last readout of the focal plane and archives the data collected as of the stop. The current implementation and the Interface itself do not insure the data collected will be scientifically useful. The `gpxAbortExp` command dumps all data collected as a result of the previous `gpxStartExp` into the bit bucket.

4.2.4. `gpxWaitForTrigger`

This command sets up the DHE's in the system to wait for a hardware or software trigger and then start an exposure. This command allows multiple PAN/DHE pairs to prepare to take data and to all start at the same instant. Because of the design of the MONSOON hardware the time skew between the actual start of data taking across multiple PAN/DHE pairs should be ± 1 nanosecond.

4.3. Status and Information Commands

These commands allow the Client systems to determine the state of the GPX system and the value of the attributes in the GPX. The standard commands are `gpxGetState`, `gpxGetAVP`, `gpxAsyncMsg` and `gpxAsyncResp`.

4.3.1. gpxGetState

The gpxGetState command returns a string giving the current state of the GPX the return string will be one of: IDLE, RESET, EXP_ACTIVE, EXP_PAUSED, ABORTING, STOPPING, DATAFR_LOCK, LASTFR_LOCK or ERROR. These various states restrict the actions that can be taken by a client system when GPX is in the state described.

4.3.2. gpxGetAVP

The gpxGetAVP command is the analog of the gpxSetAVP command. It retrieves from the GPX the value of a single attribute or a list of attributes.

4.3.3. Asynchronous Event Reporting

Under the status and information commands are the commands used to handle asynchronous events in a **GPX**. Two commands or responses within the GPX interface handle these events. First is the gpxAsyncMsg. A GPX client must be structured so that it can handle the arrival of a message that looks like a GPX command to the client and has the form

```
[cmdidxxx] gpxAsyncMsg [ <RESP> | <INFO> | <EVENT> ] "string asynchronous event cause"
```

These messages are indications of something happening in the GPX that is not the result of the previous command sent to the GPX from the client. Mentioned earlier was the possibility of an exposure complete event message. Other events that could trigger an asynchronous message are: a power glitch that resets some portion of the GPX, a change to an attribute value caused by a command from a secondary client, a failure in the DHE hardware resulting in a voltage going out of range, etc. The GPX will send the message and tag the message with <INFO> or <EVENT> if no response from the client is required or with <RESP> if the GPX wants to be sure that the message was received and will be acted on. Messages that might require a response are a reset of the GPX hardware, since it is likely that the state information in the PAN and DHE portions of the system will no longer be consistent and will require the client to run the setup procedure again to insure consistency.

The number of messages needing anything other than updating the client's local information about the GPX state is probably small but is still being determined. We have found only the reset/power glitch messages to be required in the MONSOON GPX software. This will be refined as we develop additional systems.

4.3.4. gpxAsyncResp

The gpxAsyncResp command is used to send a response required by the arrival of the gpxAsyncMsg command with a <RESP> directive. The software or hardware level that requested the response will ultimately deal with the response. As of this writing the only time this command is used is to send an async response to the DHE hardware after a reset.

4.4. Hardware Action Commands

These commands interact with specific hardware systems that may or may not be present in a particular GPX system. If the hardware does not exist, these commands always succeed i.e., return a message starting with OK. However the body of the message should say something like "gpxPower: No power control is available." or "No shutter installed" to indicate that the requested action did not take place. Each of these commands has a number of attributes that should be set by the client prior to the call or should be passed with the command message.

4.4.1. gpxPower, gpxReset

These commands control the power or reset state of the GPX systems. They both take attributes that describe the level of power or reset control to be exercised and the final power state of that level. (gpxReset is always an on then off function) Thus

"gpxPower DHE[1]=OFF PAN[1]=OFF" would turn the power to the PAN/DHE pair [1]; or

"gpxReset ALL" would reset the entire GPX system.

"gpxPower ALL=ON would power up the entire GPX.

At this time the MONSOON systems do not contain any automated power control systems. In addition the multi-level resets are not implemented.

4.4.2. gpxShutter

This command controls the current or future state of a controlled shutter. The command takes a shutterName attribute and a value of OPEN or CLOSED. If the <IMMD> directive is present the shutter is immediately opened or closed. If it is not present the shutter is opened or closed during the next exposure. The details of shutter control are still being developed as no MONSOON system as yet has a locally controlled shutter.

4.4.3. gpxShftImg

This command has been included to handle image shifting in the PAN or on the focal plane. The DLIRIM experiment run at NOAO in 1995-6 showed that we could achieve diffraction limited imaging at the Mayall 4M using this technique. The OTA device will also use the image shifting technique and this command is provided for that purpose as well. The details of the OTA devices currently under development indicate that this high-level shift command will only be used during testing and evaluation of the OTA's and the GPX client will never see the data that determines the image shifts. We are still in development on this system. (See Reference 6.)

4.5. Debugging and testing commands

The GPX provides for a number of debugging, and test commands. "gpxSimulate" puts various levels of the GPX software into a simulation mode so software can be tested while not attached to the Hardware. "gpxTestMode" will be implemented to provide a number of still to be determined diagnostic tests built into the GPX system. "gpxDebug" will set the debugging level for various portions of the GPX software to allow tracing of bugs, (what Bugs? never) and error conditions.

4.6. Meta Commands and Controlling the Interface

The last class of commands are meta commands. These do not directly implement any required functionality in the GPX system but add some refinement or additional functionality.

4.6.1. gpxHelp, gpxDump

These two commands are included to allow debugging and help information to be sent to the client software. Both commands take several directives that modify their behavior and both take a string search parameter. The command:

"gpxHelp foo",

will search the GPX command and attribute tables and return the help strings associated with all names containing the sub-string foo one per line. If the resulting response string is longer than 4095 the system will also indicate that more strings match and the client should refine the search. The command:

"gpxDump <GUI> xxx",

is a mechanism used to tell the client software about the system integrator's idea of how status data should be organized in a GUI.

4.6.2. gpxExit, gpxShutdown

These commands cause the entire GPX (gpxExit) or a single GPX process to shutdown and exit. The processes do any required house cleaning and then exit.

4.6.3. gpxConnect, gpxDisconnect, gpxSend, gpxRecv

These commands are external to the GPX interface but will be implemented by the MONSOON project in a separate libgpxClient library. They allow a client program to connect, disconnect or send commands and receive responses from a GPX system without knowing the details of sockets, ports or machine IP addresses. A database of GPX system name vs. GPX IP address and socket port will be maintained for NOAO MONSOON systems that use the MONSOON GPX.

4.6.4. gpxPass

This command provides a method to by-pass the GPX interface and protections. The command is provided for the engineers and technicians who must diagnose problems with a GPX system. The gpxPASS commands takes a series of hex values that are sent to the underlying hardware using the API established for that hardware. The commands:

```
gpxPass 0x4001FFFF or gpxPass 0x40 0x01 0xFF 0xFF,
```

would both send the four bytes "0x40 0x01 0xFF 0xFF" to the DHE hardware. In a NOAA MONSOON DHE this would request a read of board address 0x0001, register address 0xFFFF. The result of the command (is any) would be returned to the client program as a string of bytes in the order received. In this case something like

```
"OK: gpxPass got 0x40 0x01 0xFF 0xFF 0x00 0x00 0x08 0x00"
```

5. Attribute-Value Pair Convenience commands

During the development of the MONSOON PAN/DHE pair software, we discovered that it would be convenient to describe two interfaces to the outside world for a GPX. The PPX interface (NOAO ICD 5.0) is a simplified GPX interface that allows communications to the individual PAN nodes from the MONSOON Supervisor layer or from Engineering Clients. This interface includes several extra features that enable the engineer to quickly set and retrieve the values of individual attributes and to execute commands in an abbreviated form. It also restricts some of the flexibility of the GPX interface. The additional features and the restrictions both effect how the interface deals with attributes. Where the GPX can understand

```
gpxSetAVP fSample=4 coadds=16 digAvg=4;
```

the PPX will receive: (The MONSOON ppx library currently recognizes both forms.)

```
ppxSetAVP fSample=4  
ppxSetAVP coadds=16  
ppxSetAVP digAvg=4
```

However the PPX interface contains a number of convenience commands to control PAN system attributes. The PPX understands:

```
fSample 8 to be equivalent to ppxSetAVP fSample=8 and
```

```
fSample to be equivalent to ppxSetAVP fSample
```

The response from these commands has the same structure as that from the GPX commands except that the message is always of the form

```
OK: attributeName Set to attributeValue or
```

```
OK: attributeName=attributeValue or possibly
```

```
ERROR: Value 900.000 Volts out of range for VdetBias (0.000 to 1.1 volts allowed)
```

6. GPX Response Handling

The GPX interface calls for a common response pattern. The response to every command is an immediate (within 100ms) message indicating whether or not the command has been accepted or executed. In the case of commands that may take some time, i.e. gpxTestMode or gpxStartExp, the GPX may generate a later gpxAsyncMsg response. This means the GPX client program, the observation or instrument control system, must be able to deal with these asynchronous style messages.

The definition of the format of GPX response messages is still being evolved as the MONSOON software is being refined.

7. Legacy Detector Controller Commands

In our design of the GPX interface a major concern/consideration was the possibility that resources might be made available to retrofit older controller hardware with the new software. We made extensive efforts to insure that the commands available in these older systems could be duplicated in the GPX command set. The table below is a partial matrix of the legacy controller systems reviewed and the GPX commands that implement those commands. We reviewed GNAAC, wildfire, ArCon, ALICE, SDSU 1 and SDSU 2 systems as well as the NOAO 2901/2910 forth controllers that are now mostly obsolete. The result was a list of minimum commands which only controlled detectors, focal planes and the data transfer, pre-processing and archiving.

Table 1 Legacy Command Matrix		
Generic Pixel Server Command	wildFire	GEMINI IR Array Controller (GNAAC)
gpxSetMode	setup,	apply
gpxSetArrConfig	dsetup	ArSetup
gpxSetExpConfig	ask, setRdd, setMode, setb011Mode	ObsSetup
gpxSetIDPConfig	ask, setRdd, setMode, setb011Mode	ROI Setup
gpxSetAVP	setVar, SetVoltName, coAdds, Lnrs, digAvg,	apply
gpxStartExp	go	observe
gpxArmExpTrigger	<i>No Equivalent Command</i>	<i>No Equivalent Command</i>
gpxStop	stop	stop
gpxAbort	abort	abort
gpxPause	<i>No Equivalent Command</i>	pause
gpxResume	<i>No Equivalent Command</i>	continue
gpxShutter	<i>No Equivalent Command</i>	apply
gpxShftImage	<i>No Equivalent Command</i>	<i>No Equivalent Command</i>
gpxSimulate	<i>No Equivalent Command</i>	simulate
gpxTestMode	<i>No Equivalent Command</i>	test
gpxPower	shutdown, exit	shutdown
gpxReset	<i>No Equivalent Command</i>	reboot init, reset
gpxTestMode	<i>No Equivalent Command</i>	test, verify, endverify
gpxGetState	hkScreen	<i>automatic</i>
gpxGetAVP	readVar	<i>automatic</i>
gpxAsyncStatus	<i>No Equivalent Command</i>	<i>automatic</i>
gpxAsyncRespond	<i>No Equivalent Command</i>	<i>No Equivalent Command</i>
gpxPass	<i>No Equivalent Command</i>	<i>No Equivalent Command</i>
No Equivalent or out of scope	east, west, sky, guider, etc	park, guide, datum, endguide, endobserve

ACKNOWLEDGEMENTS

We would like to acknowledge the contributions made by the staff astronomers and engineers at CTIO and Kitt peak for their efforts refining the interface requirements. In addition valuable insight into the command structure was provided by members of the ASTEROIDS group and by the NOAO Data Products division. We also recognize that a great deal more work is required to refine ICD 4.0 into a document that can be accepted across observatories and organizations

REFERENCES

1. Nick C. Buchholz, Phil N. Daly, Barry M. Starr – 2002 - *NOAO Interface Control Document 4.0 - Generic Pixel Server Communications, Command/Response and Data Stream Interface Description*.
2. MONSOON Project Team – 2003 - *MONSOON Image Acquisition System (Pixel Server) - Functional and Performance Requirements Document (FPRD) – Initial Draft*.
3. N. C. Buchholz, P. N. Daly – 2003 – *MONSOON Software PDR - Requirements, Software Architecture & Design* – power point presentation.
4. N. C. Buchholz, G. Chisholm, P. N. Daly, P. Ruckle – 2004 - *Interface Control Document 1.0 - Data Handling System Interface - Status and Data Stream Transfers - Draft*
5. P. N. Daly, N. C. Buchholz and P. Moore– 2004 – *Automated Software Configuration in the MONSOON system* - Proc. SPIE Vol. 5496, Advanced Software, Control and Communications Systems for astronomy, Hilton Lewis, Gianni Raffi, Eds. (this volume)
6. D. Sawyer, P. Moore, G. Rahmer, and N.C. Buchholz – 2004 *Orthogonal Transfer Array Control Solutions Using the Monsoon Image Acquisition System* - Proc. SPIE Vol. 5499, Optical and Infrared Detectors for Astronomy, James Beletic, James D. Garnett Eds.