Steward Observatory
Mountain Operations
Bok 90-inch Telescope
90Prime Upgrade Project
https://github.com/so-90prime

90Prime Software and Hardware Guide

P. N. Daly

Mountain Operations, Steward Observatory[1]
933 N. Cherry Avenue, Tucson AZ 85719, U S A
(520) 621-3648, pndaly@arizona.edu

Abstract. This document describes the new software and hardware systems to support the upgraded 90Prime imager at the Bok 90-inch telescope.

Document revision date 20250917. Last author: Phil Daly (pndaly@arizona.edu).

Contents

[1] Steward Observatory is the research arm of the Department of Astronomy at the University of Arizona (UArizona). Its offices are located on the UArizona campus in Tucson, Arizona (US). Established in 1916, the first telescope and building were formally dedicated on April 23, 1923. It now operates, or is a partner in telescopes at five mountain-top locations in Arizona, one in New Mexico, one in Hawaii, and one in Chile. It has provided instruments for three different space telescopes and numerous terrestrial ones. Steward also has one of the few facilities in the world that can cast and figure the very large primary mirrors used in telescopes built in the early 21st century.

List of Figures

List of Tables

1. Quick Start

To start the system, execute the following if operating on bonsai (10.30.1.7) [1]:

> *ssh -XY primefocus@bonsai*
> *cd /home/primefocus/bokGalil*
> *source etc/bokGalil.sh $(pwd) gui* [2]

Allow the interface shown in Figure 1 to appear and press the *start* button. The last element to appear—indicating that startup is complete—is a *ds9* window Then choose an appropriate web interface as shown in Figure 2:

> `http://10.30.1.7:5905` for astronomers, or
> `http://10.30.1.7:5905/indi` for engineers.

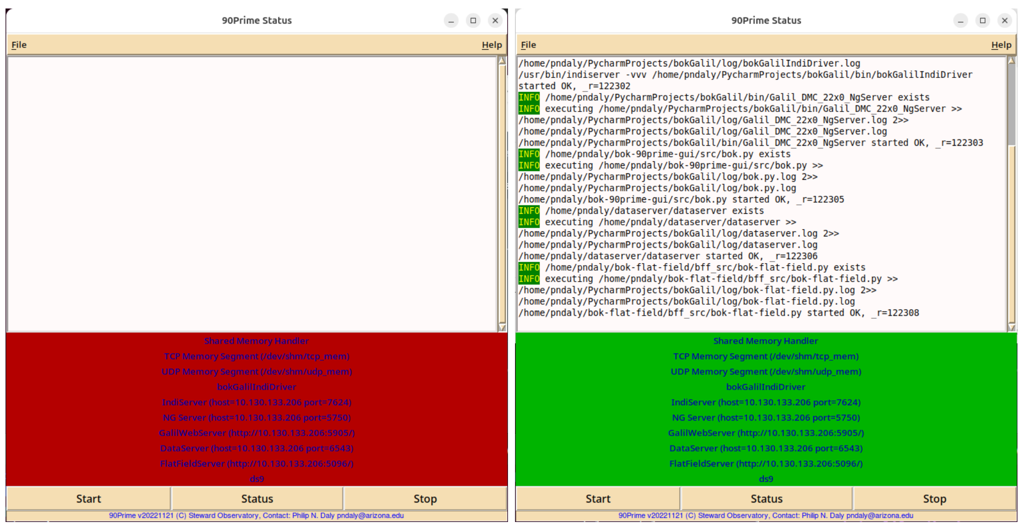

Figure 1: The 90Prime Startup and Shutdown Interface.

To stop the system, press the *stop* button. The system status can be interrogated at any time by pressing the *status* button. Figure 1 shows the system completely stopped on the left-hand side (all red) and the system fully enabled on the right-hand side (all green). If any subsystem crashes, this interface will reflect the status by changing that element from *green* to *red*. Note that the *start*, *stop* and *status* actions are global with no application to specific tasks. If a single task crashes, it is easiest to re-start the whole system (and takes a few seconds at most).

1.1. Filter File(s)

The list(s) of available filters are stored in separate files in $BOK_GALIL_DOCS:

---

[1] If running from banzai (10.30.1.8), the command is: *ssh -XY primefocus@banzai*

[2] You can also replace the *$(pwd)* syntax with the 'backward apostrophe' syntax: ` *pwd* `. The 'backward apostrophe' is often located at the top-left of the keyboard sharing the '~' key below the ESC key. However, this syntax only works in the /home/primefocus/bokGalil sub-directory. From *any* location, use the full command syntax: *source /home/primefocus/bokGalil/etc/bokGalil.sh /home/primefocus/bokGalil gui*.

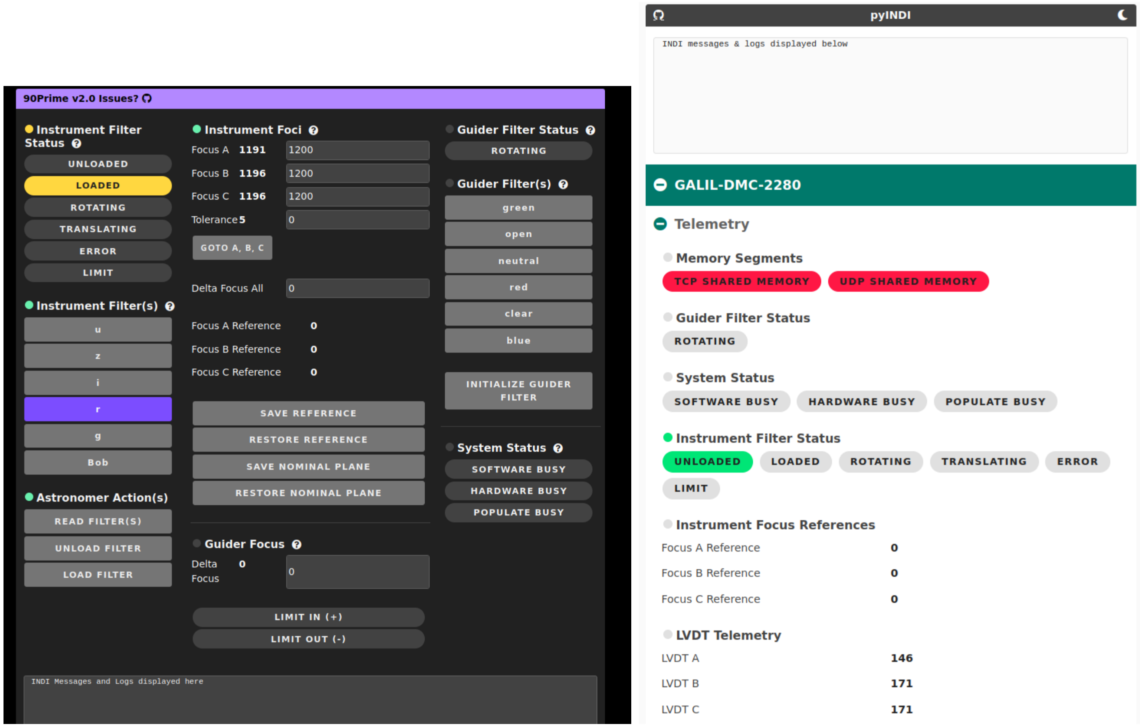

Figure 2: The 90Prime Astronomer (left) and Engineer (right, only partially shown) GUIs.

bok_gfilters.txt This file lists the available guider wheel filters;

bok_ifilters.txt This file lists the available instrument wheel filters and corresponds to the file in /home/primefocus/90prime/galil/filters.txt on BART;

bok_sfilters.txt This file lists the available guider wheel filters and should be synchronized with bok_gfilters.txt above.

These files can be edited as required. After editing, to check that the file(s) are usable by the system, execute:

cd $BOK_GALIL_BIN
./bok_read_filters -f$BOK_GALIL_DOCS/bok_gfilters.txt
./bok_read_filters -f$BOK_GALIL_DOCS/bok_ifilters.txt
./bok_read_filters -f$BOK_GALIL_DOCS/bok_sfilters.txt

If these files are changed whilst the system is running, the system *must* be re-started for them to take effect.

1.2. /nfs/data/primefocus/*yyyymmdd*

A cronjob runs every day, just after local noon, to create a new data directory in /nfs/data/primefocus/*yyyymmdd* where *yyyymmdd* is the date at the start of the observing night (*i.e. viz.,* local time). The observer or engineer will have to (manually) enter this directory in AzCam.

Transferring Data Downtown The *primefocus* accounts on both bonsai or banzai have ssh-keys registered using the *bok* account on beast.as.arizona.edu . Some command(s) to transfer this data are:

ssh bokbeast mkdir data/yyyymmdd
ssh bokbeast chown -R bok:users data/yyyymmdd
rsync -avz /nfs/data/primefocus/yyyymmdd bokbeast:data/
ssh bokbeast chown -R bok:users data/yyyymmdd

where *yyyymmdd* is the observation date associated with the data directory shown in section 1.2. on page 5. The last 2 commands should be used regularly throughout the night.

1.3. Day Crew Action(s)

When the software is started, the day crew will typically initialize the system via the following action(s) accessed via the engineering interface (`http://10.30.1.7:5905/indi`) as shown in Figure 3):

POPULATE This command, once executed, allows the day crew to install various filters using the side button on the dewar;

POPULATE DONE This command completes the populate filter wheel sequence;

INITIALIZE INSTRUMENT FILTER This command reads the instrument filter wheel and must be allowed to complete;

LOAD FILTER This command will load the currently selected filter. Do this as part of setup. This should be done before handing over to astronomers;

INITIALIZE GUIDER FILTER This (optional) command reads the guider filter wheel and, if invoked, must be allowed to complete.

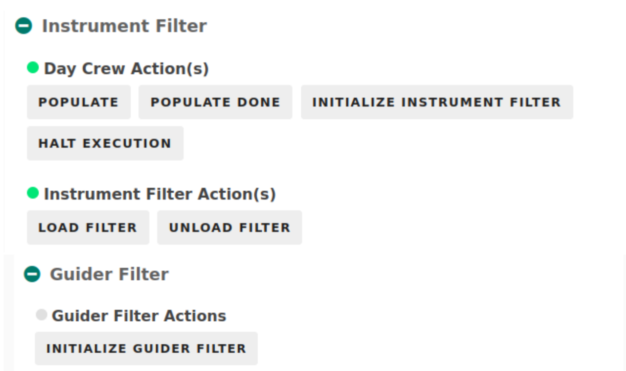

Figure 3: The Day Crew Actions Within The Engineering GUI. Make sure the radio buttons are 'green' after you are done.

1.4. Running IRAF

Once the instrument is fully connected, day crew usually take an image to visually check the output array and check counts. This can be done from b onsai or b anzai by logging into the *primefocus* account and executing:

cd iraf_docker
bash iraf yyyymmdd

or

bash iraf-community yyyymmdd

where *yyyymmdd* is the observation date associated with the data directory shown in section 1.2. on page 5. You will then have 'ccdlist', 'imexam' *etc* available within the iraf window.

1.5. Astronomer Night Action(s)

READ FILTERS This command reads the instrument filter wheel and must be allowed to complete. Do this at the start of every observing run or when the software is restarted;

LOAD FILTER This command loads the currently selected instrument filter.

UNLOAD FILTER This command unloads any previously loaded instrument filter. It is good practice to unload the filter at the end of the observing night.

1.6. Astronomer Flat Field Control

Note that as part of the 90Prime startup, a new flat field GUI is created as shown in figure 4. However, the flat field controller in the computer rack must be manually turned on for these controls to become effective. The web address for this GUI is `http://10.30.1.7:5096` if running on bonsai or `http://10.30.1.8:5096` if running on banzai .

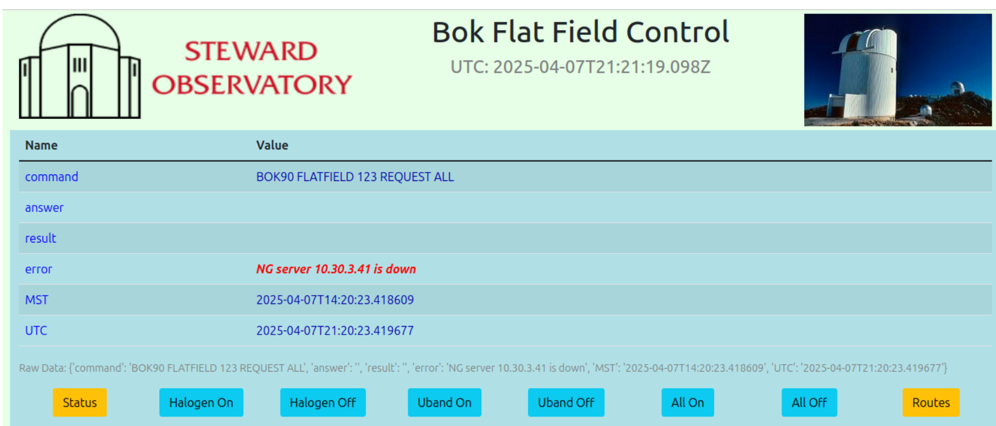

Figure 4: The Bok Flat Field Control Interface.

1.7. Astronomer Observer Log

The directory */home/primefocus/bok_observer_log* contains 2 utility programs that astronomers might find useful:

bok_observer_log.py This command produces a nightly observing log from the FITS header data within the specified path. It defaults to the observing night date in 'yyyymmdd' format. An example of the output is show in figure 5.

bok_observer_fixfits.py This command is able to fix any FITS file header issues and add new keywords to the primary HDU only.

For the latter utility, there is a JSON-like syntax required by the code. As an example we can fix the OBSERVER keyword and add a new OBSCOMM keyword in the test.fits file like so:

```
cd /home/primefocus/bok_observer_log
python3 bok_observer_fixfits.py --file=test.fits \
    --data='{"OBSERVER": "Phil Daly", "OBSCOMM": "test comment"}'
```

The OBSCOMM keyword is particularly useful since the bok_observer_log.py utility will pick it up and, if it's not empty, display it beneath the file in the output PDF. If in doubt, use the following commands to get some simple help:

```
python3 bok_observer_log.py --help
python3 bok_observer_fixfits.py --help
```

Bok Observer Log From /nfs/data/primefocus/20250326

Latitude: 31.9629° | *Longitude: -111.6004°* | *Elevation: 2071.12m* | *Altitude: 6795ft* | *Focal Length: 6.08m (19.95ft)* | *Mount: Equatorial* | *Primary: 2.3m (90-inch)*

File	UTC	Object	RA	Declination	Epoch	Filter	ExpTime (seconds)	Airmass	Azimuth (°EoN)	Elevation (°)	RotAngle (°)	LVDT A	LVDT B	LVDT C	Instrument	Binning	Camera Temp (°C)	Dewar Temp (°C)	Det Size (pixels)	File Size (Mb)
bs.ZERO.0001.fits	2025-03-27T01:13:25.006	bias	06:04:11.18	+31:47:16.5	2000.0	z	0.0	1.0	-83.9	90.0	0.0	1176.7937	1157.2618	1235.3892	90prime	1x1	-98.566	-184.851	[8160x8160]	128.34
bs.ZERO.0002.fits	2025-03-27T01:17:55.735		06:08:44.71	+31:47:26.5	2000.0	u	0.0	1.0	-83.9	90.0	0.0	1240.2723	1196.3255	1132.8471	90prime	1x1	-98.575	-184.852	[8160x8160]	128.34
bs.ZERO.0003.fits	2025-03-27T01:18:58.003		06:09:48.02	+31:47:28.8	2000.0	u	0.0	1.0	-83.9	90.0	0.0	1240.2723	1201.2086	1137.73	90prime	1x1	-98.579	-184.851	[8160x8160]	128.34
bs.ZERO.0004.fits	2025-03-27T01:22:20.420		06:13:07.96	+31:47:36.2	2000.0	u	0.0	1.0	-83.9	90.0	0.0	1250.0381	1196.3255	1137.73	90prime	1x1	-98.591	-184.853	[8160x8160]	128.34
bs.ZERO.0005.fits	2025-03-27T01:22:51.324		06:13:38.94	+31:47:37.3	2000.0	u	0.0	1.0	-83.9	90.0	0.0	1245.1552	1196.3255	1137.73	90prime	1x1	-98.591	-184.853	[8160x8160]	128.34
bs.ZERO.0006.fits	2025-03-27T01:23:22.174		06:14:09.93	+31:47:38.5	2000.0	u	0.0	1.0	-83.9	90.0	0.0	1245.1552	1196.3255	1132.8471	90prime	1x1	-98.591	-184.853	[8160x8160]	128.34
bs.ZERO.0007.fits	2025-03-27T01:23:53.087		06:14:40.86	+31:47:39.6	2000.0	u	0.0	1.0	-83.9	90.0	0.0	1245.1552	1196.3255	1132.8471	90prime	1x1	-98.591	-184.853	[8160x8160]	128.34
bs.ZERO.0008.fits	2025-03-27T01:24:23.969		06:15:11.84	+31:47:40.7	2000.0	u	0.0	1.0	-83.9	90.0	0.0	1245.1552	1201.2086	1132.8471	90prime	1x1	-98.591	-184.853	[8160x8160]	128.34
bs.ZERO.0009.fits	2025-03-27T01:24:54.860		06:15:42.83	+31:47:41.9	2000.0	u	0.0	1.0	-83.9	90.0	0.0	1245.1552	1196.3255	1132.8471	90prime	1x1	-98.591	-184.853	[8160x8160]	128.34
bs.ZERO.0010.fits	2025-03-27T01:25:25.713		06:16:13.81	+31:47:43.0	2000.0	u	0.0	1.0	-83.9	90.0	0.0	1245.1552	1196.3255	1132.8471	90prime	1x1	-98.591	-184.853	[8160x8160]	128.34

Figure 5: The Bok Observer Log Output.

2. Emergency Shutdown

Normally, the computer rack is *always on* but there are times when an emergency shutdown is warranted. Even in this case, the KVM-switch (kvm), network switch (bokqnap) and UPS (bokups) are left powered up. The important hardware to shutdown cleanly are the NAS (boknas) and 2 computers (bonsai , banzai). The following is the standard operating procedure in this circumstance.

2.1. Shutdown the NAS

1. Login to either bonsai or bonsai .
2. Start a web-browser.
3. Navigate to http://10.30.1.9:5000 ... the screen should appear looking like the LHS of Figure 6.
4. Login as *mtnops* but the *usual* password should have the 'd' character capitalized to 'D'!

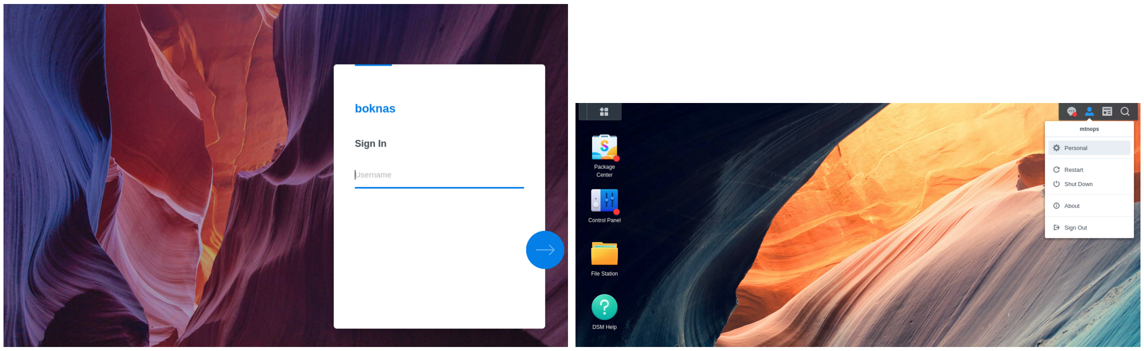

Figure 6: Shutting Down the NAS (boknas).

5. The screen should change to looking like the RHS of Figure 6.
6. Locate the person icon on the top right and click to see the drop-down menu.
7. Click on 'Shut Down'.

Allow a minute or two for the NAS to shutdown before proceeding to shutdown the computers.

2.2. Shutdown the Computers

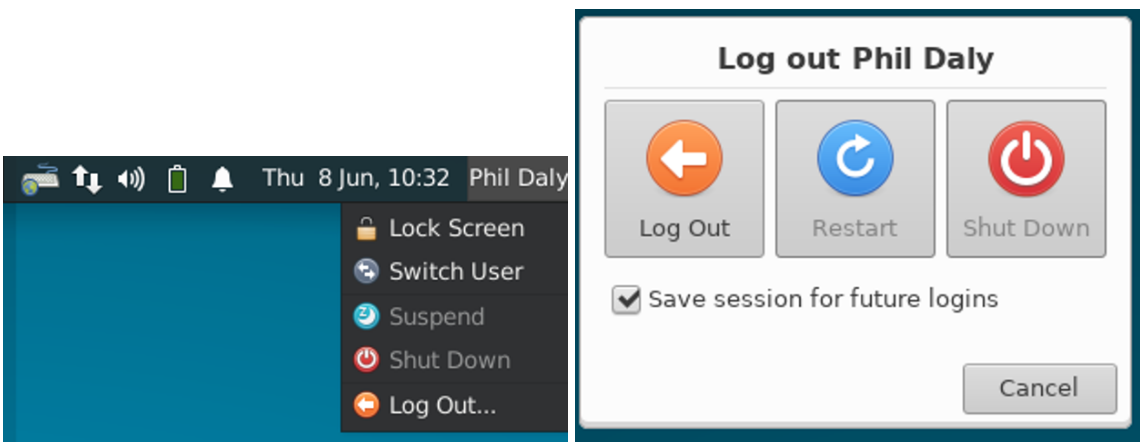

Figure 7: Shutting Down the Computers (bonsai , banzai).

The procedure for shutting down the computers is the same for both. Here we use bonsai as the example. The order is unimportant.

1. Login to either bonsai as *mtnops* with the usual password.
2. Click on the username (which should say 'mtnops') in the top right of the screen to bring up the menu.
3. Click on 'Shut Down'.
4. The pop-up window shown in Figure 7 should appear so click on 'Shut Down'.
5. Repeat for banzai .

If you have difficulty using the GUI, a power off can be executed from the command line using 'sudo poweroff' (and you will be prompted for the mtnops password again).

3. Hardware

A graphical representation of the hardware rack, installed in the 2^{nd} floor office is shown in Figure 8. The basic concept is for an ALWAYS ON system that can withstand the usual Kitt Peak lightning activity so the system is protected by a *Brickwall* surge protector on a 20A circuit. The rack is further protected by an APC 2200VA UPS. The CPU(s) are redundant inasmuch as, if bonsai fails, banzai can take over and vice-versa. It is imperative, therefore, that infrastructure software on these machines is kept synchronized. We use GitHub repositories for the Steward code base and well-documented procedures for obtaining the support software.

bonsai A Lancelot 1983-T 1U server from www.aslab.com. The specification is an Intel Xeon Silver 4210R 2.4GHz (10 cores), 6x8Gb DDR4-2933 Memory (48Gb), 2 x Samsung 970 Evo Plus 2Tb SDD (RAID1), Intel X710-DA2 2-port PCIe x8 2 x SFP+ (10 Gbps), Ubuntu 20.04 LTS (ask for 22.x), 3 years support. The system was later upgraded to Ubuntu 22.04.2 LTS. This machine has a standard *primefocus* account for running the software and a sudo-enabled *mtnops* account for privileged actions. bonsai is the primary *data acquisition* system and banzai the secondary.

banzai A Lancelot 1983-T 1U server from www.aslab.com. The specification is an Intel Xeon Silver 4210R 2.4GHz (10 cores), 6x8Gb DDR4-2933 Memory (48Gb), 2 x Samsung 970 Evo Plus 2Tb SDD (RAID1), Intel X710-DA2 2-port PCIe x8 2 x SFP+ (10 Gbps), Ubuntu 20.04 LTS (ask for 22.x), 3 years support. The system was later upgraded to Ubuntu 22.04.2 LTS. This machine has a standard *primefocus* account for running the software and a sudo-enabled *mtnops* account for privileged actions. banzai is the primary *data reduction* system and bonsai the secondary.

boknas A Synology RS1221RP+ chassis with 8x16Tb Seagate IronWolfPro+ mechanical drives. These are configured as a single 96Tb of SHR-2/RAID6 array NFS-mounted on bonsai and banzai as /nfs/data. The Synology chassis was upgraded with (a total of) 32Gb of RAM and the E10M20-T1 Ethernet / M2 combo card. This card supports the SNV3510-800 (800Gb) SSD cache and the 10Gbps network connection. The management account is *mtnops* with the usual password but with the 'd' capitalized! The management interface can be reached at `http://10.30.1.9:5000/#/signin/password`.

bokqnap A QNAP QSW-M1208-C 10Gbps (dedicated and managed) Ethernet switch. The CPUs and NAS are attached via the 10 Gbps network for disk access and via the 1 Gbps network for regular network access. The management account is *admin* with the standard *mtnops* password. The management interface can be reached at `http://10.30.0.6/#/login`.

bokups An APC 2200VA SMT2200RM2UC UPS that has enough capacity to support the whole rack. Note that this is attached to the network but has yet to be configured for automatic graceful shutdown after a given time has elapsed on battery power. Note that this may be referred to as *bokrackups* in /etc/hosts.

bokkvm An eKL VGA KVM Switch 8 Port 8x2 which supports Keyboard, Mouse, Audio, USB (although we use it in 'dumb mode'). It has an IR remote control.

4. Software

The software is centered around the *Instrument Neutral Distributed Interface*[1,2,3] and the *gclib* library provided by Galil[4]. An architecture diagram is shown in Figure 9. The software can built and run on any machine that supports the following:

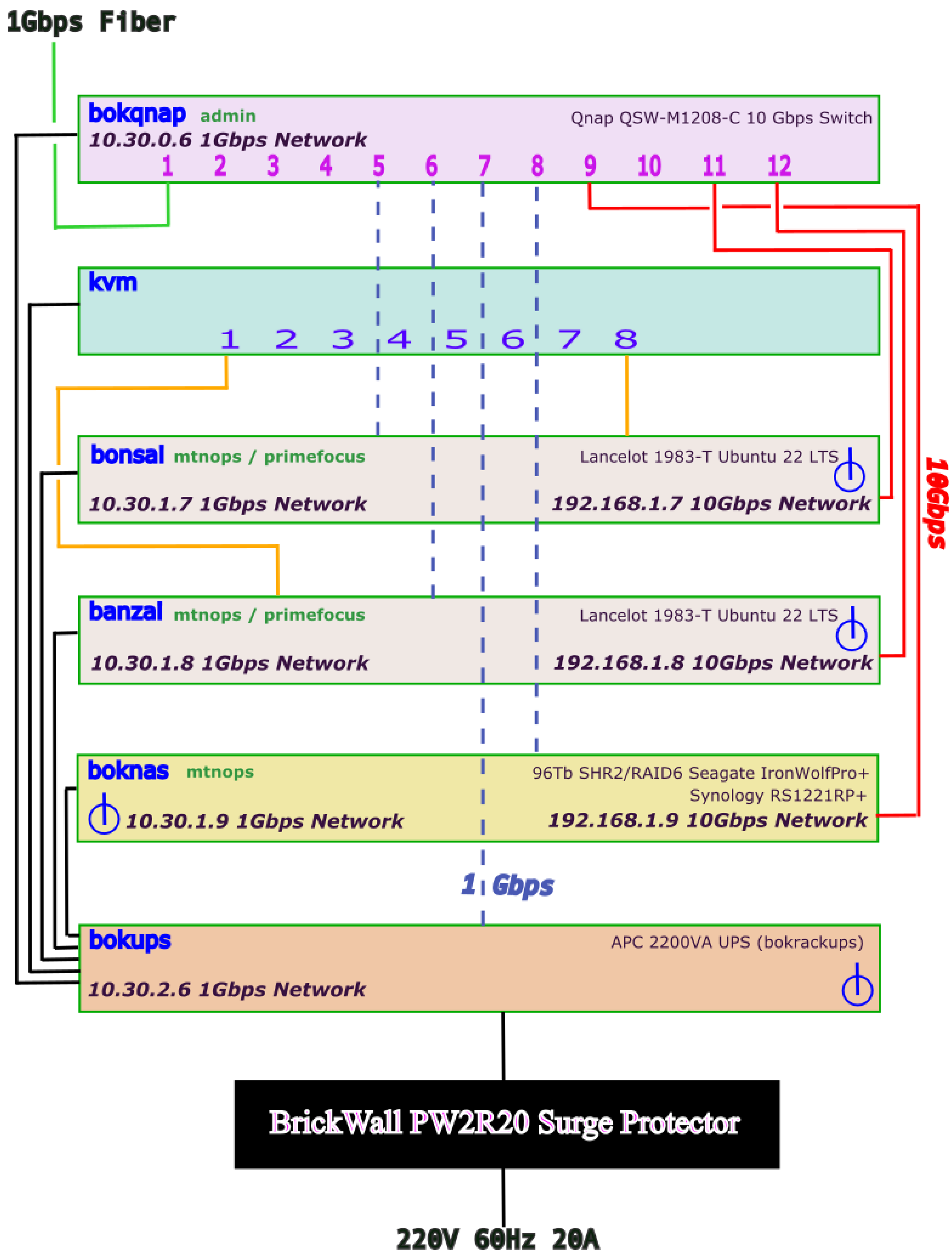

Figure 8: The 90Prime System Rack.

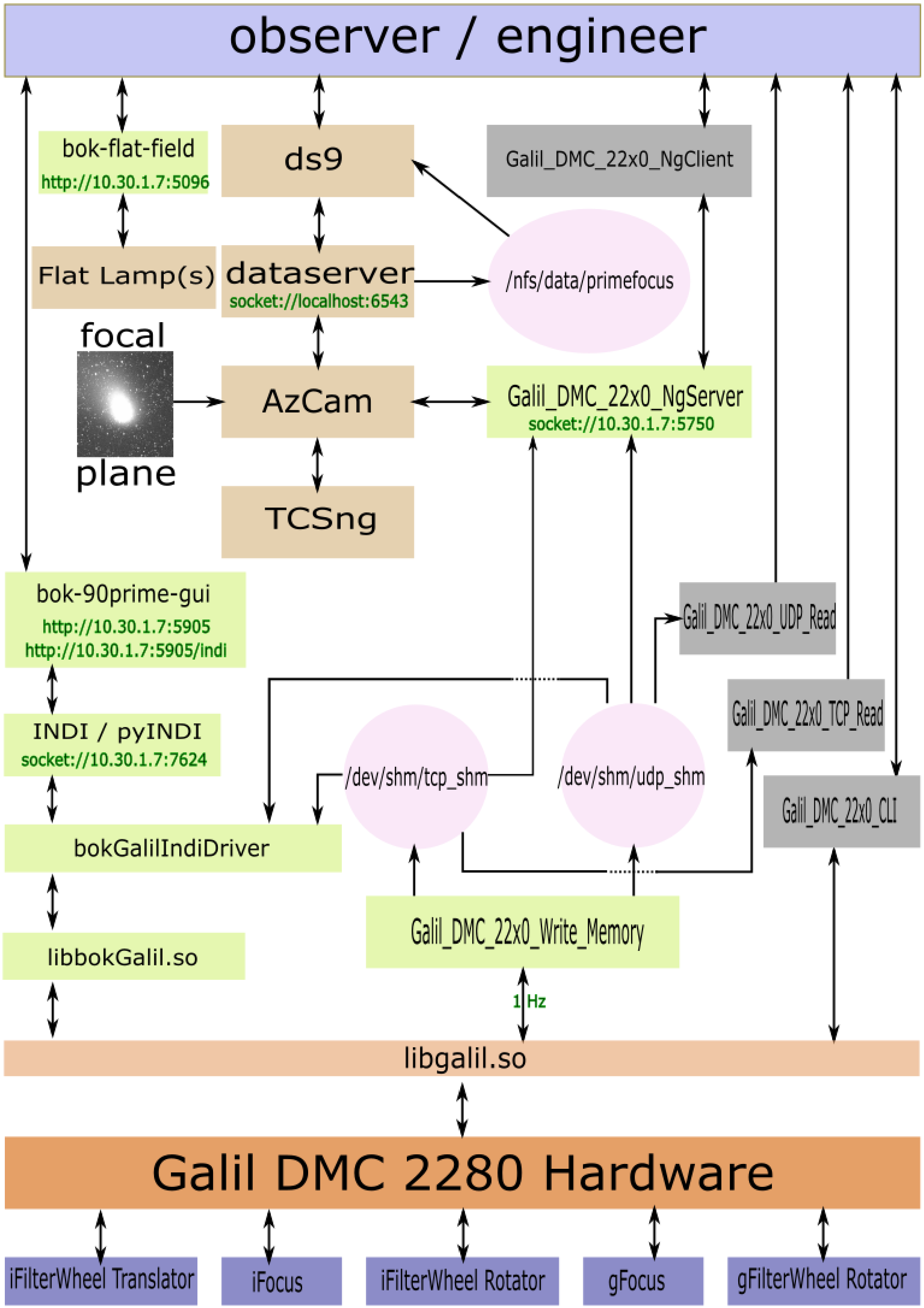

Figure 9: The 90Prime Software Architecture.

- Ubuntu 22.04 LTS (or later)
- INDI infrastructure code (www.indilib.org)
- gclib (www.galil.com)
- Python 3.8 (or later)
- gcc 11.3.0 (or later)

Note that this software only supports /dev/shm shared memory under Unix. Other repositories required to support operations are:

dataserver https://github.com/so-mops/dataserver.git

bok-90prime-gui https://github.com/so-90prime/bok-90prime-gui.git

pyINDI https://github.com/MMTObservatory/pyINDI.git

4.1. Required Components

In Figure 9 the minimum set of components required to run the system are:

INDI / pyINDI is the infrastructure software. The *indiserver* can be installed following the instructions in Appendix A. The pyINDI software from the MMT is installed direct from a GitHub repository as part of the *bok-90prime-gui* installation.

libgalil.so This is the software library delivered by the vendor. It can be installed following the instructions in Appendix B.

libbokGalil.so is the support library written by the author of this document to abstract the communications with the device.

bokGalilIndiDriver is the INDI driver for the Galil.

Galil_DMC_22x0_Write_Memory is an independent program running at 1 Hz that communicates with the hardware and delivers key telemetry to shared memory segment(s) /dev/shm/tcp_shm and /dev/shm/udp_shm.

Galil_DMC_22x0_NgServer is an NG protocol server typically used by the AzCam software to retrieve data for FITS headers. The supported NG protocol commands and requests are documented in Appendix C.

4.2. Optional Components

Galil_DMC_22x0_NgClient is *not* required for normal operations but may be run to assist in debugging.

Galil_DMC_22x0_CLI is *not* required for normal operations but may be run to assist in testing and debugging.

Galil_DMC_22x0_TCP_Read is *not* required for normal operations but may be run to assist in debugging.

Galil_DMC_22x0_UDP_Read is *not* required for normal operations but may be run to assist in debugging.

4.3. Building The Software

The software can be installed directly from the GitHub repository. The following instructions assume that the *indiserver* has been built (see Appendix A) and *gclib* installed (see Appendix B).

```
git clone https://github.com/so-90prime/bokGalil.git
cd bokGalil
mkdir lib log
source etc/bokGalil.sh $(pwd) load
cd $BOK_GALIL_HOME
sudo python3 -m pip install -r requirements.txt
make -f ./test_galil.make
```

Note that $BOK_GALIL_TCL/bokParams.bonsai.txt and $BOK_GALIL_TCL/bokParams.banzai.txt should already exist but these should be checked for consistency with values reported in Table 1. Make the appropriate system:

```
cd $BOK_GALIL_TCL
make bonsai
```

Note that $BOK_GALIL_SRC/__hosts__.bonsai.h and $BOK_GALIL_SRC/__hosts__.banzai.h should already exist but these should be checked for consistency with values reported in Table 1. The __hosts__.py file is automatically created. Make the appropriate system:

```
cd $BOK_GALIL_SRC
make bonsai
```

Re-build this document (as required):

```
cd $BOK_GALIL_TEX
make all
```

4.4. Test

test_galil The easiest way to test the *gclib* installation is to execute the test code:

```
cd $BOK_GALIL_HOME
./test_galil -h || ./test_galil -b0 || ./test_galil -b1
```

telnet A further test of the connectivity is to use the standard *telnet* utility:

```
telnet 10.30.3.31
```

then execute the LV; command. Data should appear. Use CTRL-] to escape to the telnet prompt and enter QUIT.

XEphem Connect with XEphem − > View − > Sky View − > Telescope − > INDI panel − > Connect.

Web Browser Connect either to the astronomer interface (`http://10.30.1.7:5905`) or the engineering interface (`http://10.30.1.7:5905/indi`) as shown in Figure 2.

Table 1: bokGalil Configuration Variable(s).

Variable	Description	bonsai	banzai
GALIL	Galil DMC 2280 Hardware	192.168.0.100	192.168.0.100
BOK_GALIL_CMD_BOK	Galil DMC 2280 Hardware	10.30.3.31	10.30.3.31
BOK_GALIL_CMD_LAB	Galil DMC 2280 Spare	192.168.0.100	192.168.0.100
BOK_INSTRUMENT	Instrument	90Prime	90Prime
BOK_INDI_ADDR	IndiServer Address	10.30.1.7	10.30.1.8
BOK_INDI_PORT	IndiServer Port	7624	7624
BOK_NG_ADDR	NG Server Address	10.30.1.7	10.30.1.8
BOK_NG_PORT	NG Server Port	5750	5750
BOK_TCP_ADDR	Galil TCP Command Address	10.30.3.31	10.30.3.31
BOK_TCP_PORT	Galil TCP Command Port	23	23
BOK_UDP_ADDR	Galil UDP Command Address	10.30.1.7	10.30.1.8
BOK_UDP_PORT	Galil UDP Command Port	5078	5078
BOK_WEB_ADDR	pyINDI Website Address	10.30.1.7	10.30.1.8
BOK_WEB_PORT	pyINDI Website Port	5905	5905
BOK_WEB_REPO	pyINDI Website Repository	/home/primefocus/bok-90prime-gui	/home/primefocus/bok-90prime-gui
BOK_DATA_ADDR	DataServer Address	10.30.1.7	10.30.1.8
BOK_DATA_PORT	DataServer Port	6543	6543
BOK_DATA_REPO	DataServer Repository	/home/primefocus/dataserver	/home/primefocus/dataserver
BOK_FF_REPO	FlatField Website Repository	/home/primefocus/bok-flat-field	/home/primefocus/bok-flat-field
BOK_FF_ADDR	FlatField Website Address	10.30.1.7	10.30.1.8
BOK_FF_PORT	FlatField Website Port	5096	5096

4.5. Debugging

Log files are written to $BOK_GALIL_LOG. The Galil_DMC_22x0_CLI interface may also be run: at the command prompt enter '?' for options. Any Galil supported command may be sent to the hardware if the Galil_DMC_22x0_CLI program is invoked with the *-o* (override) option!

References

[1] https://en.wikipedia.org/wiki/Instrument_Neutral_Distributed_Interface.

[2] https://www.indilib.org.

[3] http://www.clearskyinstitute.com/INDI/INDI.pdf.

[4] https://www.galil.com/sw/pub/all/doc/gclib/html/ubuntu.html.

A INDI Installation

Execute the following commands (as root):

apt update
apt-get install -y git cdbs dkms cmake swig fxload libev-dev libgps-dev libgsl-dev libraw-dev
apt-get install -y libusb-dev zlib1g-dev libftdi-dev libgsl0-dev libjpeg-dev libkrb5-dev
apt-get install -y libnova-dev libtiff-dev libfftw3-dev librtlsdr-dev libcfitsio-dev
apt-get install -y libgphoto2-dev build-essential libusb-1.0-0-dev libdc1394-dev
apt-get install -y libboost-regex-dev libcurl4-gnutls-dev libtheora-dev libxml2-utils

Build the software (as root):

rm -rf /usr/local/IndiProjects
mkdir -p /usr/local/IndiProjects
cd /usr/local/IndiProjects
git clone https://github.com/indilib/indi.git
mkdir -p /usr/local/IndiProjects/build/indi-core
cd /usr/local/IndiProjects/build/indi-core
cmake -DCMAKE_BUILD_TYPE=Debug /usr/local/IndiProjects/indi
make -j4
make install

Optionally, install the Python client (as root):

apt-get install -y python3-pip
python3 -m pip install –upgrade pip
python3 -m pip install pyindi-client

B gclib Installation

Execute the following commands (as root). First, get and install the key:

wget https://www.galil.com/sw/pub/all/crypto/GALIL-GPG-KEY-E29D0E4B.asc
mv GALIL-GPG-KEY-E29D0E4B.asc /etc/apt/trusted.gpg.d/

Second, update the repository list:

curl -O https://www.galil.com/sw/pub/ubuntu/22.04/galil.list
mv galil.list /etc/apt/sources.list.d/

Finally, install the software:

apt update
apt remove gclib gcapsd
apt install gclib gcapsd

C NG Protocol Commands and Requests

The software supports the standard NG protocol syntax:

```
<telescope> <instrument> <cmd-id> <COMMAND||REQUEST> <extra-information>
```

If <cmd-id> is set to SIMULATE, no hardware is accessed and dummy response(s) are returned! Commands and requests are case insensitive.

All *commands*, return one of the following responses:

On success bok 90prime <cmd-id> OK

On failure bok 90prime <cmd-id> ERROR <reason>

All *requests*, return one of the following responses:

On success bok 90prime <cmd-id> OK <returned-data-values>

On failure bok 90prime <cmd-id> ERROR <reason>

C1. Supported Command(s)

bok 90prime <cmd-id> command exit — client informs server it's shutting down.

bok 90prime <cmd-id> command gfilter init — client commands server to initialize guider filter wheel.

bok 90prime <cmd-id> command gfilter name <str> — client commands server to change guider filter to given name.

bok 90prime <cmd-id> command gfilter number <int> — client commands server to change guider filter to given number.

bok 90prime <cmd-id> command gfocus delta <float> — client commands server to change guider focus to given value.

bok 90prime <cmd-id> command ifilter init — client commands server to initialize instrument filter wheel.

bok 90prime <cmd-id> command ifilter name <str> — client commands server to change instrument filter to given name.

bok 90prime <cmd-id> command ifilter number <int> — client commands server to change instrument filter to given number.

bok 90prime <cmd-id> command ifilter load — client commands server to insert current filter into beam.

bok 90prime <cmd-id> command ifilter unload — client commands server to remove current filter from beam.

bok 90prime <cmd-id> command ifocus a <float> b <float> c <float> t <float> — client commands server to change instrument focus in all 3 axes by separate amounts within tolerance.

bok 90prime <cmd-id> command ifocusall delta <float> t <float> — client commands server to change instrument focus in all 3 axes by the same amount within tolerance

bok 90prime <cmd-id> command lvdt a <float> b <float> c <float> t <float> — client commands server to change instrument LVDTs in all 3 axes by separate amounts within tolerance.

bok 90prime <cmd-id> command lvdtall <float> t <float> — client commands server to change instrument LVDTs in all 3 axes by the same amount within tolerance.

bok 90prime <cmd-id> command test — client commands server to test communication path.

bok 90prime <cmd-id> command hx — client commands server to halt execution in the galil controller.

C2. Supported Request(s)

bok 90prime <cmd-id> request encoders — client requests encoder values. An example response might be 'BOK 90PRIME <CMD-ID> OK A=-0.355 B=1.443 C=0.345'.

bok 90prime <cmd-id> request gfilter — client requests server to report current guider filter. An example response might be 'BOK 90PRIME <CMD-ID> OK GFILTN=4:RED ROTATING=FALSE'.

bok 90prime <cmd-id> request gfilters — client requests server to report guider filters. An example response might be 'BOK 90PRIME <CMD-ID> OK 1=1:GREEN 2=2:OPEN 3=3:NEUTRAL 4=4:RED 5=5:OPEN 6=6:BLUE'.

bok 90prime <cmd-id> request gfocus — client requests server to report guider focus. An example response might be 'BOK 90PRIME <CMD-ID> OK GFOCUS=-0.355'.

bok 90prime <cmd-id> request ifilter — client requests server to report current instrument filter. An example response might be 'BOK 90PRIME <CMD-ID> OK FILTVAL=18:BOB INBEAM=TRUE ROTATING=FALSE TRANSLATING=FALSE ERRFILT=0 FILTTSC=3'.

bok 90prime <cmd-id> request ifilters — client requests server to report instrument filters. An example response might be 'BOK 90PRIME <CMD-ID> OK 0=18:BOB 1=2:G 2=3:R 3=4:I 4=5:Z 5=6:U'.

bok 90prime <cmd-id> request ifocus — client requests server to report instrument focus. An example response might be 'BOK 90PRIME <CMD-ID> OK A=-0.355 B=1.443 C=0.345'.