

Общее описание первой версии системы управления телескопами комплекса «АстроМ»

Емельянов Эдуард Владимирович

Лаборатория физики оптических транзисторов САО РАН

2022-07-20

Содержание

1 Состав комплекса	1
2 Состав предварительной системы управления телескопами	3
2.1 Метеорологические параметры	3
2.2 Управление телескопом	4
2.3 Управление монтировкой «10Micron»	4
2.4 Управление куполами	6
2.5 Управление питанием аппаратуры	7
2.6 Управление узлами фотометра	8
2.7 Вспомогательные утилиты	10
3 Юстировка полярной оси и коррекция системы наведения	11

1 Состав комплекса

В данный момент комплекс представлен тремя независимыми телескопами на экваториальных монтировках в составе:

1. Панорамный купол «Baader AllSky Dome 4.5-m» с метеостанцией (не работают датчики влажности и атмосферного давления); телескоп «Astrosib RC-500» на монтировке «10Micron GM4000HPS»; компьютер системы управления `robotel1.sao.ru`; управляемая розетка `robosock1.sao.ru` на основе одноплатного компьютера «Orange Pi zero»; фотометр в составе ПЗС «Fli ProLine PL16801», турели с фильтрами B, V, R и r', фокусера «FLI Atlas». Фотометр установлен в ньютоновском фокусе телескопа. На внутренней стенке купола размещен осветитель «плоского поля» размером 60 × 60 см.
2. Панорамный купол «Astrosib ASD-4.5»; телескоп «Astrosib RC-500» на монтировке «10Micron GM4000HPS»; компьютер системы управления `robotel2.sao.ru`; управляемая розетка `robosock2.sao.ru` на основе одноплатного компьютера «Orange Pi zero»; фотометр в составе ПЗС «Fli ProLine PL16801» и турели с фильтрами B, V, R и r'. Фотометр установлен в кассегреновском фокусе телескопа (используется штатный фокусер вторичного зеркала телескопа).

3. Панорамный купол «Astrosib ASD-4.5»; телескоп «Astrosib RC-500» на вилочной монтировке «Астросиб BM-700» (управляемой контроллером «SiderealServo-II»); ПЗС «FLI ProLine PL16801» для технических нужд установлена в фокусе Кассегрена.

Одноплатники управляемых розеток функционируют на основе ОС Armbian Linux (на основе Ubuntu), все остальные компьютеры (управляющие телескопами и сервер) — на основе ОС Gentoo Linux.

Кроме того, комплекс оснащен (неработающей) системой пожарной и охранной сигнализации; системой видеонаблюдения, подключенной к (неработающему) серверу-видеорегистратору, а также опрашиваемой установленным на roboserv.sao.ru ПО Zoneminder, позволяющим осуществить мониторинг всех видеокамер.

Характеристики телескопов «Astrosib RC-500» с ПЗС FLI.

- Среднее качество изображения: $1.^{\circ}5 \div 2.^{\circ}$.
- Масштаб в прямом фокусе: $1.^{\circ}35$ на пиксель 9 мкм.
- Поле зрения в прямом фокусе: $1.^{\circ}53 \times 1.^{\circ}55$.
- Масштаб в фокусе Кассегрена: $0.^{\prime\prime}46$ на пиксель.
- Поле зрения в фокусе Кассегрена: $32.^{\prime}1 \times 31.^{\prime}9$.
- Вращение поля вследствие ошибок юстировки полярной оси: $< 20.^{\prime\prime}/\text{мин.}$
- Фотометрическая точность¹: до $0.^m1$ для звезд $19.^m$.

Грубый механизм юстировки монтировок 10Micron не позволяет выставить полярную ось достаточно точно, в результате чего наблюдается небольшое остаточное вращение поля.

Монтировки 10Micron имеют встроенную систему коррекции наведения (СКН), однако, список точек с данными коррекции ограничен всего лишь 100 парами координат. Без этой системы точность наведения составляет не лучше $1..2^{\circ}$, а точность сопровождения объекта хуже $15.^{\prime\prime}$ за десять минут. В режиме включеной СКН точность наведения составляет не хуже $15.^{\prime\prime}$, точность сопровождения — не хуже $3.^{\prime\prime}$ за 10 минут экспозиции.

Обнаружен значительный (до 0.1 мм за ночь — $5.^{\prime\prime}5!$) температурный дрейф фокуса, что приводит к необходимости выполнения дополнительных фокусировок в течение наблюдательной ночи.

Телескоп №1 введен в режим регулярной опытной эксплуатации в феврале 2020 г., №2 эксплуатируется в опытном режиме с января 2021 г (изначально фотометр так же, как и на телескопе №1 был установлен в ньютоновском фокусе, с апреля 2022 г перенесен в фокус Кассегрена). Телескоп №3 нуждается в разработке специальной системы управления (т.к. имеет примитивный контроллер монтировки, в отличие от №1 и №2), кроме того, не завершены монтажные и пусконаладочные работы в башне этого телескопа (в т.ч. прокладка линий электропитания и связи; установка системы осушения воздуха в подкупольном пространстве).

Также в состав комплекса входит сервер обработки данных roboserv.sao.ru, дисковый накопитель robonas.sao.ru и второстепенное оборудование. Для удобства проведения примитивных операций (включить питание, открыть купол и т.д.) в куполе телескопа №1 установлен wifi-роутер.

Для организации наблюдений в полуавтоматическом режиме (решение о возможности начать наблюдения, а также начальную фокусировку проводит человек) разработан ряд демонов с сетевым доступом, позволяющих осуществлять управление узлами комплекса. Дальнейшее выполнение наблюдений производится по сценарию, например, в виде bash-скрипта.

¹2021GCN.29277....1M, 2021ATel14459....1V

2 Состав предварительной системы управления телескопами

2.1 Метеорологические параметры

Для получения доступа к метеорологическим параметром на `robotel1.sao.ru` работает утилита `weatherdaemon`², ее параметры:

- P, --pidfile=arg** полный путь к PID-файлу (по умолчанию `/tmp/weatherdaemon.pid`);
- b, --baudrate=arg** скорость работы с последовательным портом, к которому подключена локальная метеостанция (по умолчанию: 9600);
- d, --device=arg** полный путь к устройству последовательного порта;
- e, --emulation** эмуляция последовательного порта для тестов (регулярно выдается одно и то же сообщение);
- h, --help** отображение справки;
- l, --logfile=arg** файл для сохранения логов (по умолчанию отсутствует);
- p, --port=arg** номер порта сокета, в который будут выдаваться полученные метеоданные (по умолчанию 12345);
- v, --verb** каждый **-t** увеличивает уровень детализации сообщений и логгирования.

Утилита один раз в 10 секунд опрашивает локальную метеостанцию, откуда берется информация о температуре воздуха, скорости ветра и наличии осадков. Также опрашивается СУ БТА на предмет метеоданных. При подключении к сокету на заданном порту выдается информация (далее она повторяется каждые 10 секунд — сразу после получения новых данных с местной метеостанции), например:

```
Rain=1
Clouds=0.0
Exttemp=10.7
Wind=1.4
BTAEtttemp=12.1
BTAPres=597.2
BTAWind=0.9
BTAHumid=98.9
Time=1658239138
```

Поля с префиксом «БТА» содержат информацию с метеостанции БТА. Форсировать получение данных можно, передав любой символ в сокет (при этом будет отображен последний пакет информации). Поле «Time» содержит UNIX-time последнего измерения. Поле «Clouds» содержит данные датчика, измеряющего условную температуру неба (чем больше его значение, тем лучше качество изображения; оптимальным для начала наблюдений является значение не меньше 2500).

²https://github.com/eddyem/small_tel/tree/master/Daemons/weatherdaemon

Для упрощения использования метеоданных в файлах-сценариях разработан bash-скрипт `chkweather`³, который производит разбор полученных из сокета (при помощи утилиты `netcat`) данных и, помимо того, что выводит их в `stdout`, завершается с кодом ошибки в случае, если погодные условия не позволяют проводить наблюдения (Clouds меньше 1700, наличие осадков, либо ветер свыше 15 м/с). В случае, если в течение пяти минут скрипт возвраща-ет ошибку, необходимо прекратить наблюдения, припарковать телескоп и закрыть створки телескопа и купола.

2.2 Управление телескопом

Полностью управление узлами телескопа еще не реализовано (т.е. нет возможности полу-чить величину температуры его узлов, включить вентиляторы охлаждения главного зеркала или нагрев вторичного), есть лишь возможность работать со створками телескопа (открыть/закрыть), для чего на телескопе № 1 работает утилита `teldaemon`⁴. Ее аргументы:

- d, --device=arg** полный путь к устройству последовательного порта;
- e, --echo** «эхо» команды пользователя;
- h, --help** отображение справки;
- l, --logfile=arg** файл для сохранения логов (по умолчанию отсутствует);
- p, --port=arg** номер порта сокета, по которому будут приниматься команды (по умолча-нию 4444);
- t, --terminal** работать в режиме терминала, передавая все пользовательские данные в порт и выдавая обратно ответ.

Посыпать команды (`open`, `close` и `status`) можно как напрямую в сокет при помощи `netcat`, так и из адресной строки браузера или запросом `curl`, например:

```
curl localhost:4444/status
```

На телескопе № 2 управление производится иначе: т.к. он работает в кассегреновском фокусе, задавать положение фокуса и открывать/закрывать створки необходимо прямыми командами в порт. Для этой цели используются скрипты `Chfocus`⁵ (задание положения фокусера; кроме того, этот скрипт заполняет файл `FOCUS.hdr` в локальной директории данными для FITS-заголовка с информацией о текущем отсчете фокусера), `OpenScope`⁶ и `CloseScope`⁷ (открыть и закрыть створки телескопа).

2.3 Управление монтировкой «10Micron»

Сетевой взаимосвязью пользователя с монтировкой является утилита `stellariumdaemon`⁸, имеющая следующие аргументы:

³https://github.com/eddyem/small_tel/blob/master/Daemons/weatherdaemon/chkweather

⁴https://github.com/eddyem/small_tel/tree/master/Daemons/teldaemon

⁵https://github.com/eddyem/small_tel/blob/master/Daemons/astrosib/Chfocus

⁶https://github.com/eddyem/small_tel/blob/master/Daemons/astrosib/OpenScope

⁷https://github.com/eddyem/small_tel/blob/master/Daemons/astrosib/CloseScope

⁸https://github.com/eddyem/small_tel/tree/master/Daemons/10micron_stellarium

- D, --dbgport=arg** сетевой порт, позволяющий работать напрямую с терминалом управления монтировкой (для ввода команд напрямую);
- P, --pidfile=arg** PID-файл (по умолчанию `/tmp/stellariumdaemon.pid`);
- W, --wname=arg** имя или IP-адрес компьютера, на котором запущен «погодный демон»;
- d, --device=arg** полный путь к устройству последовательного порта;
- e, --emulation** работа в режиме эмуляции (только для клиента Stellarium);
- h, --help** отображение справки;
- l, --logfile=arg** файл для сохранения логов (по умолчанию отсутствует);
- p, --port=arg** номер порта сокета, по которому будут приниматься команды от Stellarium (по умолчанию 10000);
- w, --wport=arg** номер порта сервера «погодного демона».

Утилита позволяет задавать положение телескопа из виртуального планетария Stellarium, а также вводить любые команды напрямую. Например, так можно навести телескоп на точку $A = 90^\circ, Z = 45^\circ$:

```
#!/bin/bash

A="90:00:00"
H="45:00:00"

function sendcmd(){
    echo $1 | nc 192.168.70.33 10001 -q10
}

sendcmd ":Sz${A}#"
sendcmd ":Sa${H}#"
sendcmd ":MS#"
while true; do
ANS=$(sendcmd ":Gstat#")
echo $ANS
[ $ANS == "0#" ] && break
sleep 2
done
```

Скрипт `GetCoords`⁹ позволяет получить текущие координаты телескопа. Также для этих целей служит утилита `send_coords`¹⁰, позволяющая работать с монтировкой через интерфейс Stellarium. Ее аргументы:

- H, --host=arg** имя или IP-адрес хоста, где запущен демон «10Micron» (по умолчанию: `localhost`);

⁹https://github.com/eddyem/small_tel/blob/master/Auxiliary_utils/bash_scripts/GetCoords

¹⁰https://github.com/eddyem/small_tel/tree/master/Daemons/send_coordinates

- P, --port=arg** порт сокета на этом хосте (умолчание: 10000);
- d, --dec=arg** склонение цели (DD:MM:SS.SS);
- h, --help** отображение справки;
- m, --monitor** мониторинг координат (утилита не завершается, пока пользователь не прервёт ее);
- q, --quiet** подавлять любой вывод в терминал;
- r, --ra=arg** прямое восхождение цели (HH:MM:SS.SS).

2.4 Управление куполами

Управление куполами так же реализовано в виде сетевого сервиса. Купол «Baader» управляется сервисом `domedaemon`¹¹. Вспомогательная задача этого сервиса — не дать куполу закрыться во время наблюдений по таймауту `watchdog'a`. Аргументы:

- d, --device=arg** полный путь к устройству последовательного порта;
- e, --echo** «эхо» команды пользователя;
- h, --help** отображение справки;
- l, --logfile=arg** файл для сохранения логов (по умолчанию отсутствует);
- p, --port=arg** номер порта сокета, по которому будут приниматься команды (по умолчанию 55555);
- t, --terminal** работать в режиме терминала, передавая все пользовательские данные впорт и выдавая обратно ответ.

Команды управления аналогичны командам телескопа.

Подобным же образом устроен сервис управления куполом «Astrosib»¹². При работе с этим куполом отмечена проблема: сбой управляющего компьютера в случае ручного (при помощи кнопок на панели контроллера) перемещения створок купола в промежуточное положение между полностью открытым и полностью закрытым состоянием (например, частично приоткрыть купол). Этот сбой чреват аварией с физическим разрушением купола, поэтому крайне не рекомендуется пользоваться кнопками на панели.

При эксплуатации куполов «Baader» и «Astrosib» в зимнее время отмечена проблема в них обоих: в случае, если створка купола упирается в лед или плотный снег так, что привод двигателя входит в состояние ошибки по превышению рабочего тока, сброс ошибки возможен исключительно вручную путем снятия и подачи питания снова. При этом с большой вероятностью у купола «Astrosib» возможна вышеуказанный поломка. Купол «Baader» же можно оснастить магнитным пускателем, разрывающим цепи питания приводов в случае такой проблемы.

Для купола «Astrosib» требуется полная переработка системы управления (приводы допускают управление по RS-485, в т.ч. диагностику и сброс ошибок) с полной заменой управляющей логики. Возможно, установка дешевых многооборотных абсолютных энкодеров на валы двигателей упростит и значительно обезопасит процесс управления куполом.

¹¹https://github.com/eddyem/small_tel/tree/master/Daemons/domedaemon

¹²https://github.com/eddyem/small_tel/tree/master/Daemons/domedaemon-astrosib

2.5 Управление питанием аппаратуры

На телескопах № 1 и № 2 в роли управляемых розеток выступают одноплатные компьютеры Orange Pi Zero, к которым подключены релейно-симисторные модули на основе микроконтроллера STM8S003¹³. Помимо двух силовых коммутационных узлов модули имеют один выход на р-канальном MOSFET с открытым стоком, два выхода с открытым стоком на п-канальных MOSFET, один логический вход без развязки, один вход с развязкой через оптопару и два выхода с развязкой через оптопары. Один из выходов обоих контроллеров питания (выход 1) используется для включения системы управления монтировкой «10Micron».

Модуль управления питанием подключается к управляющему одноплатнику по USB через преобразователь USB–UART, скорость 9800 бод, 8N1. Протокол управления текстовый. Началом является символ двоеточия, концом — символ решетки. Между ними помещается один из управляющих символов:

- A** выключить симистор канала 0;
- B** выключить симистор канала 1;
- c/C** проверить входы 0/1;
- i/I** отобразить амплитуду тока в канале 0;
- k/K** включить/выключить ключ P0;
- l/L** включить/выключить ключ N0;
- m/M** включить/выключить ключ N1;
- n/o** активировать выход 0/1;
- N/O** деактивировать выход 0/1;
- s/S** отобразить текущее состояние входов и выходов;
- y/Y** включить/выключить реле 0;
- z/Z** включить/выключить реле 1.

Команды отсылаются в модули напрямую из bash-скриптов¹⁴. На управляющих телескопами машинах настроен ssh-вход на «розетки» по ключам, что позволяет при помощи простейших сценариев выполнять включение-выключение¹⁵.

На телескопе № 1 также установлена подсветка подкупольного пространства и светильник «плоского поля», подключенные через двухканальный USB-модуль реле. Модуль управляетя посредством утилиты `relay_manage`¹⁶. На нулевом канале установлено освещение подкупольного, а на первом — «плоского поля». Аргументы командной строки:

- d, --device=arg** полный путь к файлу устройства (если к компьютеру подключено несколько реле, udev-скрипт создает в директории `/dev` файлы `relayX`, где X – номер устройства);

¹³https://github.com/eddyem/STM8_samples/tree/master/220controlled_socket

¹⁴https://github.com/eddyem/small_tel/tree/master/Daemons/socket

¹⁵например, https://github.com/eddyem/small_tel/blob/master/Auxiliary_utils/bash_scripts/HWoff

¹⁶https://github.com/eddyem/eddys_snippets/tree/master/USBrelay

- h, --help** отобразить справку;
- q, --quiet** не отображать в терминал информацию;
- r, --reset=arg** выключить реле с заданным номером;
- s, --set=arg** включить реле с заданным номером.

Аргументы **-s** и **-r** могут встречаться в любом количестве (т.е. сразу можно включить/выключить несколько каналов).

2.6 Управление узлами фотометра

Для управления всеми узлами FLI используется утилита **fli_control**¹⁷ (рекомендуется заменить ее на более современную **ccd_capture**¹⁸). Аргументы:

- 8, --8bit** работа в восьмибитном режиме (если доступна);
- A, --author=arg** значение поля AUTHOR FITS-заголовка;
- D, --display** отобразить полученное изображение в псевдоцветах в окне OpenGL;
- F, --fast** быстрый режим считывания;
- I, --instrument=arg** значение поля INSTRUME FITS-заголовка;
- N, --obsname=arg** значение поля OBSERVER FITS-заголовка;
- O, --object=arg** значение поля OBJECT FITS-заголовка;
- P, --prog-id=arg** значение поля PROGRAM FITS-заголовка;
- V, --verbose** увеличение уровня подробности сообщений;
- Y, --objtype=arg** значение поля OBJTYPE FITS-заголовка;
- a, --addsteps=arg** относительное перемещение фокусера на заданное количество шагов;
- c, --conf-ioport=arg** конфигурация порта I/O (1 – выход, 0 – вход для каждого бита);
- d, --dark** снимать «темновые» кадры;
- f, --nflushes=arg** сделать N засветок матрицы перед экспозицией (не работает для PL16801);
- g, --goto=arg** переместить фокусер в заданную абсолютную позицию (количество шагов);
- h, --hbin=arg** горизонтальный биннинг;
- i, --get-ioport** получить состояние порта I/O;
- n, --nframes=arg** количество кадров в серии;

¹⁷https://github.com/eddyem/mytakepic/tree/master/fli_control

¹⁸https://github.com/eddyem/CCD_Capture

-p, --pause=arg пауза (в секундах) между кадрами серии;

-r, --addrec внести в FITS-заголовок записи из указанного файла (таких файлов может быть несколько);

-s, --set-ioport=arg установить значение порта I/O;

-t, --set-temp=arg установить температуру чипа ПЗС;

-v, --vbin=arg вертикальный биннинг;

-w, --wheel-set=arg установить заданную позицию колеса турели фильтров (начиная с нуля);

-x, --exptime=arg установить время экспозиции (миллисекунд);

--X0=arg а также --X1, --Y0 и --Y1 – установить координаты части фрейма (в случае значения -1 выбирается минимальная или максимальная величина, включая область оверскана);

--async асинхронное (без ожидания окончания) движение шагового двигателя турели/фокусера;

--close-shutter закрыть затвор;

--help отобразить справку;

--open-shutter открыть затвор;

--rewrite перезаписать выходной файл, если он существует (иначе будет выдано сообщение об ошибке и файл записан не будет);

--shutter-on-high затвор откроется и экспозиция начнется при наличии высокого сигнала на контакте 5 порта I/O;

--shutter-on-low затвор откроется и экспозиция начнется при наличии низкого сигнала на контакте 5 порта I/O.

При проведении первых технических наблюдений было замечено, что данные модели ПЗС не оснащены ИК-засветкой, гарантирующей повторяемость результатов измерений. В итоге на снимках появлялись «духи» (особенно от ярких объектов). ПЗС были модифицированы: зазор между покровным стеклом ПЗС и затвором увеличен до 0.8 мм, а по периметру крышки вокруг затвора установлено четыре ИК-светодиода (850 нм). Управление светодиодами подключено к каналу 8 порта I/O ПЗС. Для организации preflash в данном случае необходимо использовать сценарий:

```
#!/bin/bash

echo "Wait, pre-flashing is in process"
fli_control -c8 >/dev/null
fli_control -s0 >/dev/null
fli_control -x2000 -d -F
```

```
fli_control -s8 >/dev/null
fli_control -c0 >/dev/null
fli_control -F -x1 -v32 -h32 -n2 -d >/dev/null
fli_control -x1 -v32 -h32 -d >/dev/null
fli_control -x1 -v2 -h2 -d
```

В данном случае выполняется настройка контакта 8 порта I/O на выход, значение контакта устанавливается в нуль (что включает светодиоды, чьи аноды подтянуты к питанию через токозадающий резистор), далее выполняется двухсекундная экспозиция с закрытым затвором в режиме быстрого считывания. После чего светодиоды отключаются и порт переконфигурируется обратно в режим входа. Последующие четыре чтения в разных режимах полностью снимают «духи» от засветки, выводя уровень сигнала на средний уровень bias. Рекомендуется делать preflash перед каждым научным кадром и кадром «плоского поля». В случае съемки темновых и bias'ов этого не требуется.

Отмечено, что при работе в быстром режиме считывания следующий кадр в обычном режиме получается испорченным. Поэтому при переходе между разными скоростями считывания необходимо делать тестовый промежуточный кадр. Для ускорения считывания можно использовать биннинг (-v32 -h32).

2.7 Вспомогательные утилиты

Для запуска наблюдений рекомендуется использовать скрипт `STARTobs`¹⁹, который включает питание аппаратуры, открывает створки телескопа и купола, устанавливает температуру светоприемника фотометра в -45°C . Скрипт `STOPobs`²⁰ выполняет указанные действия в обратном порядке, закрывая створки и отключая питание. Обратите внимание на то, что этот скрипт не паркует телескоп перед отключением питания, запарковать необходимо скриптом `park_telescope`. В случае телескопа №1 парковочным положением является наведение телескопа на осветитель «плоского поля».

При наблюдении в режиме автоматического сценария без участия человека необходимо выполнить следующую базовую последовательность действий.

1. Проверить состояние погоды (`chkweather`). Если скрипт завершен с ошибкой, запускать наблюдения нельзя.
2. Включить питание и открыть телескоп и купол (`STARTobs`).
3. В цикле проверять состояние погоды и выполнять наблюдения по программе (менять фильтры в турели, менять фокусное расстояние на поправку к фильтру, выполнять экспозицию перенаводить телескоп).
4. Проверить состояние телескопа при помощи `send_coords` (если оно не равно 0, перенавести телескоп на объект). Также, чтобы не потерять экспозиций, можно проверять командой `:Gmte#`, отправляемой в отладочный порт `stellariumdaemon` время (в минутах) до перекладки (если оно меньше времени экспозиции плюс две–три минуты, вызвать `send_coords` для перенаведения телескопа с перекладкой).
5. В случае неблагоприятных условий погоды в течение более, чем 3–5 минут, либо окончании ночи, припарковать телескоп и закрыть купол, иначе — вернуться к п. 3.
6. При необходимости выполнить съемки «плоских полей» и прочих калибровочных данных.

¹⁹https://github.com/eddyem/small_tel/blob/master/Auxiliary_utils/bash_scripts/STARTobs

²⁰https://github.com/eddyem/small_tel/blob/master/Auxiliary_utils/bash_scripts/STOPobs

7. Выполнить **STOPobs** для полного завершения наблюдений.

В случае переменчивой погоды допускается парковку телескопа и пункт 6 отложить до наступления утра, а при закрытом куполе мониторить погоду: в случае, если состояние вернется в нормальное и будет таковым свыше 15 минут, можно открыть купол и продолжить наблюдения.

Для упрощения определения времени начала и окончания наблюдений можно воспользоваться утилитой **noon**²¹. По умолчанию эта утилита отображает время прохода видимого диска Солнца через небесный меридиан (т.е. истинный солнечный полдень). Время — в секундах UNIX-time. В случае, если утилита запущена под именем **sunrise**, отображается время ближайшего восхода Солнца с пересечением заданного угла под горизонтом (аргумент утилиты). Запуск под именем **sunset** дает время захода. Без аргументов вычисляется время окончания или начала астрономических сумерек (для определения времени начала или окончания наблюдений). Помимо значения угла аргументом может быть «**standard**» (пересечение горизонта), «**civil**» — граница гражданской ночи, «**nautic**» — граница навигационной ночи или «**astro**» — граница астрономической ночи.

3 Юстировка полярной оси и коррекция системы наведения

Юстировка полярной оси производится механически при помощи ручного пульта управления телескопом. Для точной коррекции понадобится также переносной компьютер (чтобы иметь возможность выставить звезду в центр кадра). Юстировка производится итеративно путем последовательных наведений на Полярную звезду и звезду вдалеке от полюса (ближе к югу). Вначале коррекцией с пульта южная звезда устанавливается в центр кадра ПЗС, далее телескоп наводится на Полярную и при помощи шестиугольных ключей необходимо выставить ось телескопа так, чтобы звезда находилась в центре кадра. Посредством небольшого количества итераций ось выставляется с точностью до десятка угловых секунд.

Для расчета данных системы коррекции наведения (СКН) необходимо получить астрометрические поля минимум в сотне равномерно распределенных по полусфере точках. Координаты точек рассчитываются при помощи octave-функции **angdist**:

```
num_pts = 100;
indices=[0:num_pts/2-1]+0.5;
phi = acos(1 - 2*indices/num_pts);
theta = pi * (1 + sqrt(5)) * indices;
phi=pi/2-phi; % Zd -> H
lowest=20*pi/180;
idx=find(phi<lowest);
phi(idx)=[] ;
theta(idx)=[] ;

s=size(phi,2);

A=[] ; H=[] ;
```

²¹https://github.com/eddyem/small_tel/tree/master/Auxiliary_utils/Sun

```

for i = 1:s; A=[A 180-mod(theta(i), 2*pi)*180/pi]; H=[H phi(i)*180/pi]; endfor
[~,idx]=sort(A);
As=A(idx); Hs=H(idx);

f=fopen("azimuth-altitude", "w");
for i = 1:s
Az = As(i);
if (Az < 0); Az += 360.; endif
fprintf(f, "%s\t%s\n", ang(Az), ang(Hs(i)));
printf("%s\t%s\n", ang(Az), ang(90-Hs(i)));
endfor
fclose(f);

```

Она использует функцию `ang`, преобразующую угловую меру в строковый вид:

```

function s=ang(a)
% convert angle (in degr) into a string DD:MM:SS
sgn = "";
if(a < 0) sgn = "-"; a = -a; endif
d = floor(a);
a = (a-d)*60.;
m = floor(a);
a = (a-m)*60.;
s = sprintf("%s%02d:%02d:%02d", sgn, d, m, round(a));
endfunction

```

В итоге генерируется файл, который отправляется на вход скрипта, выполняющего наведения телескопа на все точки в алт-азимутальных координатах:

```

#!/bin/bash
# ./sendcmds < azimuth_altitude

function sendcmd(){
    echo $1 | nc 192.168.70.33 10001 -q10
}

badweather=0

while read A H; do
chkweather 1600 > lastweather && badweather=0 || badweather=$((badweather+1))
[ $badweather -gt 5 ] && break
echo "badweather=$badweather"
sendcmd ":Sz${A}#"
sendcmd ":Sa${H}#"
sendcmd ":MS#"
while true; do
ANS=$(sendcmd ":Gstat#")
echo $ANS
[ $ANS == "0#" ] && break
done
done

```

```

sleep 2
done
/usr/bin/fli_control -r /tmp/10micron.fitsheader -x20000 field
done

```

После запускается скрипт `solve_all`, выполняющий астрометрию:

```

#!/bin/bash
for f in *; do
solve-field --use-sextractor --no-remove-lines --uniformize 0 --radius 2 \
-p -L 1.3 -H 1.4 -u arcsecperpix $f
done
rm -f *.axy *.corr *.xyls *.match *.rdls *.solved *.wcs

```

Астрометрия производится при помощи пакета `astrometry.net`²², установленного локально (с локально развернутыми индексными файлами необходимого масштаба). В итоге получается набор файлов с суффиксом `.new`, использующихся далее. Для окончательных расчетов таблицы ошибок используется утилита `PCS_create`²³ (к сожалению, разработчики монтировки 10Micron не позаботились об автоматическом или полуавтоматическом выполнении подобных работ):

```

PCS_create -Hd --ha > PCS_list
PCS_create -HDd --ha > PCS_listHAdelta
PCS_create -t > PCS_list_10micron

```

В результате получается список ошибок в удобочитаемом человеком виде (для исключения выбросов) и в виде, согласно протоколу монтировки «10Micron». Можно скорректировать выходной файл `PCS_list_10micron`, оставив 100 объектов без выбросов погрешностей наведения (монтировка не принимает более ста значений). Далее этот файл передается в монтировку при помощи скрипта:

```

#!/bin/bash

function sendcmd(){
    echo "Send $*"
    echo "$*" | nc localhost 10001 -q10
}

sendcmd "pause"
sendcmd ":newalig#"
while read str; do
sendcmd $str
done < PCS_list_10micron
sendcmd ":endalig#"
sendcmd ":getalst#"
sendcmd "continue"

echo "Well done"

```

²²<http://astrometry.net/>

²³https://github.com/eddyem/small_tel/tree/master/Auxiliary_utils/`PCS_create`

Команды «pause» и «continue» дают понять утилите **stellariumdaemon**, что необходимо прервать или продолжить прием внешних команд наведения (для обеспечения безопасности передачи данных). Далее каждая вычисленная точка отклонений передается в монтировку и дается команда пересчитать коэффициенты СКН.