

Матобеспечение новой системы управления Цейсс-1000

[В.Шергин](#)

2014г.

Новая [система управления Цейсс-1000](#) разрабатывалась начиная с [2008-го](#) года. До [2010-го](#) выполнялись подготовительные работы в лабораторных условиях. В [2011-м](#) году проводились первые пробные наблюдения с реальным телескопом. По их результатам результатам в течение [2012-го](#) года система доводилась до эксплуатационного варианта. Система находится в штатной эксплуатации с лета [2013](#) года.

Реализован архитектурный принцип «клиент-сервер». Главная часть МО выполнена как сетевой сервер на *Java* под ОС *Linux*. Она загружается вместе с ней и работает непрерывно. Клиентские приложения, разработанные в 2010-[2014](#) годах, написаны на разных языках программирования.

Содержание:

- [Основные принципы разработки МО](#)
- [Общая структура матобеспечения Цейсс-1000](#)
- [Авторизация](#)
- [Позиционные астрономические расчеты](#)
- [Обеспечение астрономических расчетов \(*environment*\)](#)
 - [Служба времени](#)
 - [Параметры IERS](#)
 - [Метеоданные](#)
- [Угловые датчики \(энкодеры\)](#)
- [Управление двигателями телескопа](#)
- [Управление куполом](#)
- [Коррекция наведения \(*Pointing Model*\)](#)
- [Ограничения движения](#)
 - [Ограничения телескопа \(доступная область\)](#)
 - [Ограничения навесной аппаратуры \(запрещенные области\)](#)
- [Административная подсистема](#)
- [Разработка клиентских приложений](#)
 - [Java](#)
 - [JavaScript](#)
 - [Python](#)
 - [C](#)
 - [C++](#)
 - [Shell](#)
- [Служебные файлы](#)
- [Загрузка](#)
- [МО CAN-шины](#)

Основные принципы разработки МО

Разработка основной части МО выполнялась на языке *Java*. Небольшая часть (того что трудно или невозможно написать на *Java*) выполнялась на языке *C*, оформлялась в виде *shared*-библиотек и подключалась к *Java* через *JNI*-интерфейс. Это:

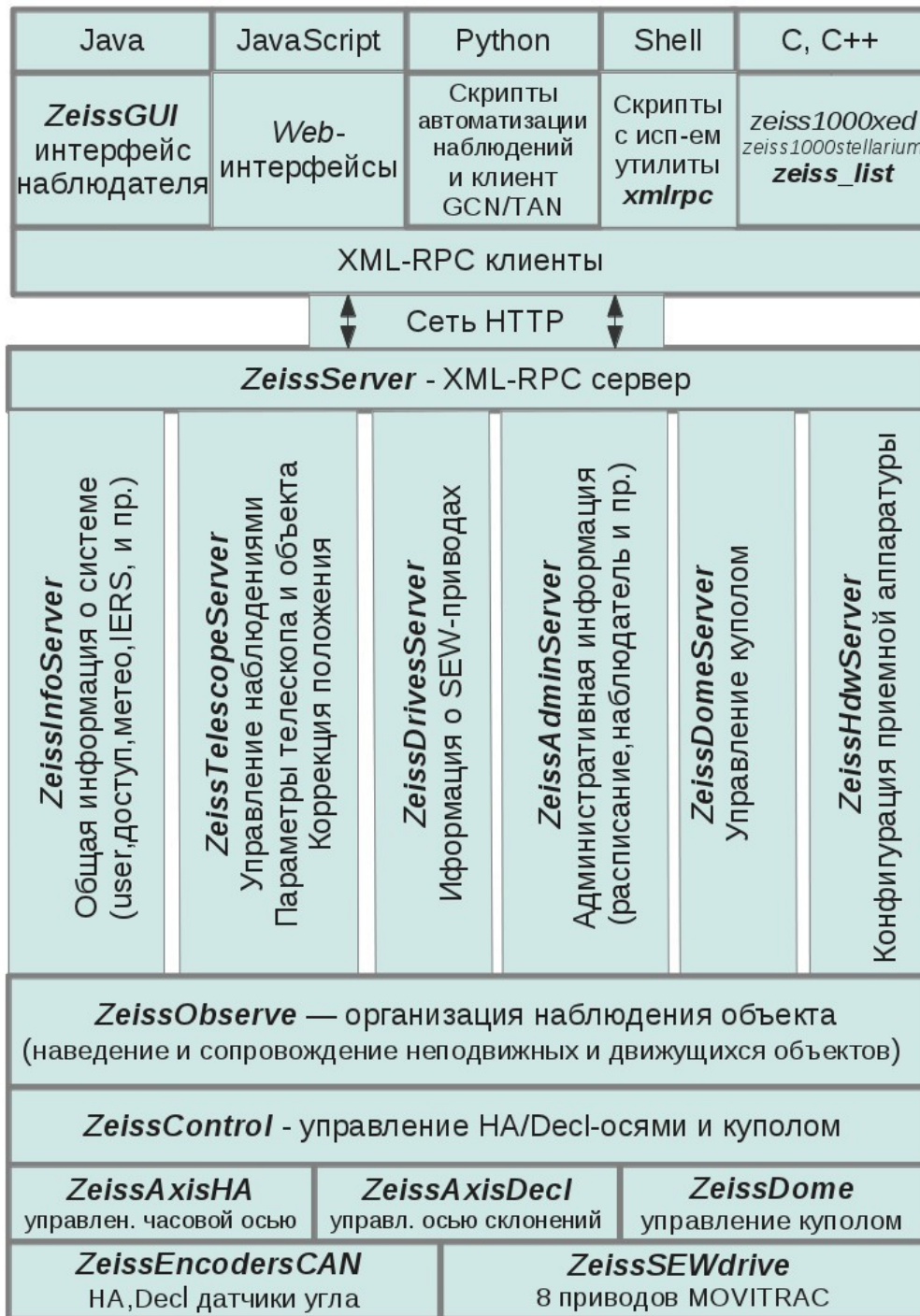
- обмен по *CAN*-шине с *SEW*-приводами и датчиками угла,
- получение метео-информации от метеослужб Цейсс-1000 и БТА,
- библиотека позиционных астрономических расчетов [*SLALIB*](#) ([*Patrick Wallace, Starlink*](#)).

Главная часть МО загружается вместе с ОС *Linux* и работает непрерывно. Она разработана как XML-RPC сервер. Для реализации базовых сервисов используется пакет *ws/xmlrpc* для *Java* из проекта *Apache*.

Протокол XML-RPC работает поверх стандартного HTTP. Это один из первых протоколов Web-сервисов. Он достаточно старый и распространенный. Поддержка его имеется для большинства языков и ОС. Это позволяет разрабатывать клиентские приложения на разных языках в разных ОС. Реально использовались *Java*, *JavaScript*, *Python*, *C*, *C++*. Все это может работать с разных компьютеров в локальной сети.

Общая структура матобеспечения Цейсс-1000

Серверная часть системы управления имеет иерархическую структуру. Ее можно разделить на уровни по принципу: компоненты верхних уровней создают и используют те что ниже. Представленные на схеме названия ее частей это имена реальных *Java*-классов. Разумеется на схеме представлены не все, а только самые важные компоненты системы управления.



ZeissServer

Главная часть сервера. Разработан на основе пакета *ws/xmlrpc* из проекта *Apache*. При старте подключает (регистрирует) предметные суб-серверы как RPC-объекты, Их методы при этом становятся удаленно вызываемыми процедурами в этих объектах.

Во время работы принимает по протоколу HTTP XML-RPC-запросы, проверяет авторизацию и вызывает методы подключенных суб-серверов.

RPC-обращения из клиентов выглядят как *ИмяОбъекта.Метод(Параметры)*.
В сервере также реализовано и стандартное *Instrospection API* позволяющее через объект *system* получать всю информацию об RPC-объектах и их методах.

ZeissInfoServer

Предоставляет общую информацию о сервере системы, которая доступна без задания имени пользователя. Эта информация представлена на [вкладке Server](#) интерфейса *ZeissGUI*.

ZeissTelescopeServer

Реализует все команды управления телескопом как при наблюдениях, так и в ручных режимах. Использует модули *ZeissObserv* и *ZeissControl*. Предоставляет всю доступную в них информацию.

Note: на самом деле создается две иерархии объектов управления наблюдением начиная с *ZeissObserve*. Одна для реального телескопа с реальными энкодерами и двигателями, другая для режима симуляции, она точно такая же только на нижнем уровне к ней подключаются объекты-симуляторы двигателей и энкодеров. Переключение между ними выполняется из клиента через данный сервер. При включении симуляции в нее копируются данные из реального управления.

ZeissDrivesServer

Предоставляет информацию из модулей *ZeissSEWdrive* о состоянии SEW-приводов телескопа и купола. Эта информация представлена на [вкладке Drives](#) интерфейса *ZeissGUI*.

ZeissAdminServer

Реализация [административной подсистемы](#).

ZeissDomeServer

Реализует все команды управления поворотом купола Цейсс-1000. Использует модуль *ZeissDome* (через *ZeissControl*). Предоставляет всю доступную в нем информацию, которая отображается на [вкладке Dome](#) интерфейса *ZeissGUI*.

ZeissHdwServer

Этот сервер выполняет все команды связанные с параметрами аппаратуры регистрации (методами наблюдений): смена аппаратуры, считывание/спасение файлов конфигурации, выдача и изменение параметров.

ZeissObserv

Организация наблюдения объекта т.е. наведения и сопровождения. Для управления телескопом использует *ZeissControl*.

ZeissControl

Управление движением разных режимах (наведение, сопровождение, ручное управление) телескопа через модули движения по осям *ZeissAxisHA* и *ZeissAxisDecl*. Реальное положение телескопа получает от *ZeissEncodersCAN*. Здесь же реализована система ограничений движения: горизонтом, куполом, северной колонной и запрещенными областями из-за навесной аппаратуры.

ZeissAxisHA* и *ZeissAxisDecl

Реализует управление вращением осей *HA* и *Dec* в разных режимах с помощью 4-х экземпляров *ZeissSEWdrive* для оси *HA* и 3-х для оси *Dec*.

ZeissDome

Реализует управление куполом в разных режимах используя экземпляр *ZeissSEWdrive* №9. В режиме согласования купола с телескопом непрерывно решает задачу преобразования *HA,Dec* телескопа -> азимут купола, учитывающую нецентральное положение трубы под куполом

ZeissEncodersCAN

Считывание показаний угловых датчиков *HA* и *Dec* с использованием протокола *CANopen*.

ZeissSEWdrive

8 экземпляров этого класса управляют 7-ю приводами *MOVITRAC* фирмы *SEW* на телескопе и еще одним №9 на куполе (№8 зарезервирован для привода фокусировки). Реализуется обмен пакетами по *CAN*-шине в соответствии с собственным протоколом фирмы *SEW*.

Авторизация

Поскольку протокол *XML-RPC* работает поверх стандартного *HTTP*, для сетевой идентификации пользователей используется стандартная авторизация заложенная в протоколе *HTTP*. Со стороны системы управления она подключена к основному серверу *ZeissServer*. Проверяется каждый запрос. Результатом идентификации является уровень доступа от 0 до 5. Нуль означает отсутствие идентификации.

Каждому пользователю приписывается уровень доступа от 1 до 5 который определяет какие действия он может выполнять. Уровень 1 — только просмотр состояния, 2 — плюс ввод координат, 3 — простой наблюдатель, 4 — опытный наблюдатель, 5 — администратор системы.

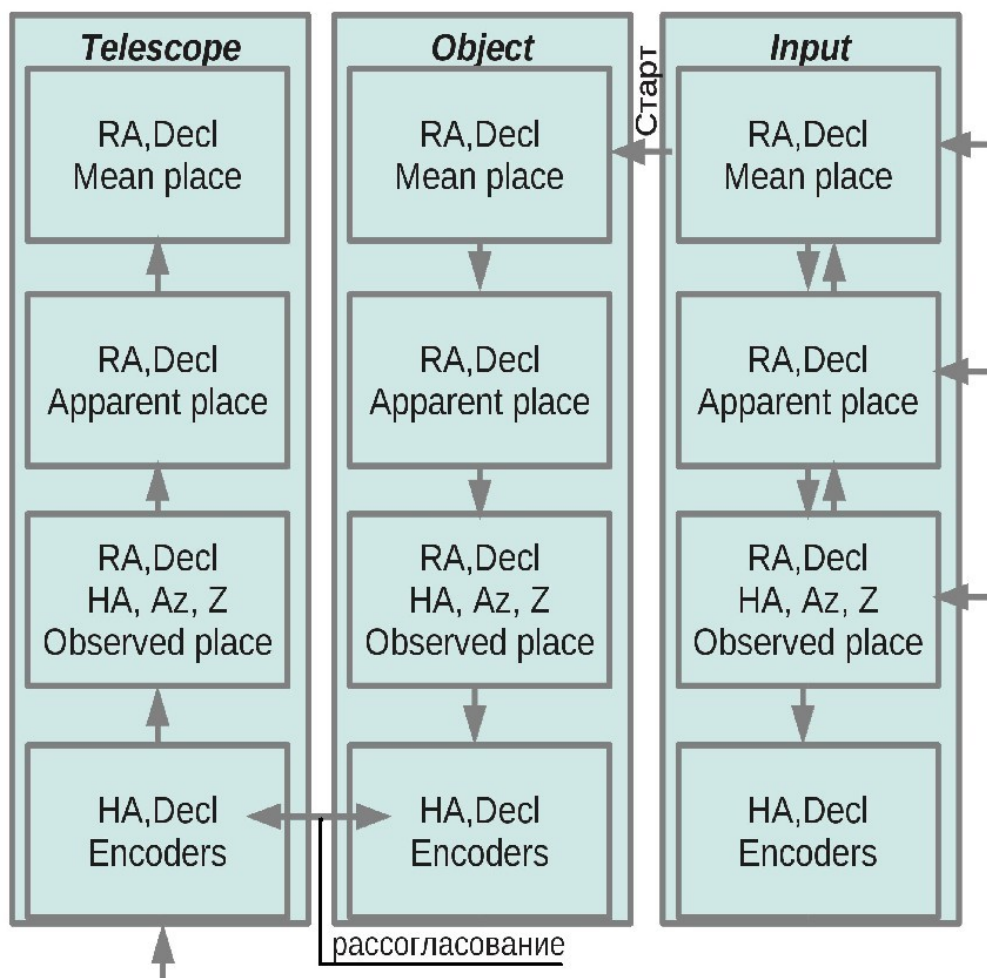
Все зарегистрированные у *ZeissServer* суб-серверы могут проверить уровень доступа обращения к ним, и либо выполнить команду, либо отказать в обслуживании. Клиент должен знать и в каждом обращении пересылать имя и пароль пользователя. Обычно они вписываются в *URL* обращения к *XML-RPC*-серверу (*http://user:passwd@hostname:8088*).

Позиционные астрономические расчеты

В классе организации наблюдения *ZeissObserv* для расчета и хранения позиционных данных

используются три одинаковых представителя класса *ScopeObject*: *Telescope*, *Object* и *Input*. Каждый представляет все типы координатных данных и расчетов:

- **Mean Place** — среднее место, т.е. RA,Dec эпохи каталога J2000 или B1950;
- **Apparent Place** — видимое место, т.е. RA,Dec пересчитанные к текущей дате;
- **Observed Place** — наблюдаемое место, т.е. RA,Dec, а также часовой угол(HA), азимут(Az) и зенитное расстояние(Z) для трубы телескопа;
- **Encoders** — показания датчиков угла на осях HA и Dec трубы телескопа.



Предназначение этих структур (*Java*-объектов):

- *Telescope* — представляет текущее состояние телескопа. На входе - углы энкодеров. Порядок расчетов: Энкодеры -> Наблюдаемое место -> Видимое место -> Эпоха каталога.
- *Object* — представляет наблюдаемый объект. Расчеты производятся в обратном порядке: от Эпохи каталога -> ... -> до углов Энкодеров.
- *Input* — служит для предварительного ввода параметров объекта. Его содержимое переписывается в *Object* при старте наведения. Данные в него могут вводиться на разных уровнях, при этом остальные рассчитываются.

Сравнение *Object* и *Telescope* выполняется на уровне энкодеров и дает текущее рассогласование.

Все координатные расчеты выполняются с использованием библиотеки [SLALIB](#) (автор [Patrick Wallace](#)), которая оформлена в виде *shared*-библиотеки и подключена к *Java*-программам сервера через *JNI*-интерфейс.

Обеспечение астрономических расчетов (environment)

Служба времени

Своей службы времени на БТА и Цейсс-1000 нет. Используется системное время ОС *Linux*. Это UTC которое синхронизируется системным сервером *ntpd* по протоколу NTP с сервером на ННП. Нормального NTP-сервера с GPS на ВНП пока что нет.

Пересчет UTC в среднее (*mean*) и истинное (*apparent*) звездное время происходит во время координатных расчетов в *Telescope*, *Object* и *Input* через функции библиотеки *SLALIB*.

Параметры IERS

Параметры Службы Вращения Земли (IERS) сервер берет из файла *ser7.dat*. Это поправка к UTC и положение земного полюса. На БТА этот файл должны периодически скачивать операторы. На Цейсс-1000 их нет. Поэтому при старте сервер проверяет дату этого файла и если он старше недели автоматом запускает скачивание его с сайта IERS. В будущем, если сервер будет работать месяцами без перезагрузки, эту проверку надо будет сделать периодической.

Метеоданные

Метеоданные необходимы для расчета рефракции. Сервер сначала пытается извлечь их сайта метеостанции Цейсс-1000 (Davis Vantage Pro) на этом же компьютере. При неудаче пытается получить их по стандартному сетевому интерфейсу АСУ БТА.

Также как и на БТА учет метеоданных блокируется во время ведения объекта чтобы не получить помех от их изменения.

Угловые датчики (энкодеры)



На телескопе установлены угловые датчики фирмы *FRABA* подключенные прямо к CAN-шине. Протокол доступа — *CANopen* (DS301V4.0 Profile:DS406, C6-class). Для обмена с ними разработан *Java*-класс *ZeissEncodersCAN* имитирующий пакеты протокола *CANopen*.

Датчики многооборотные. Один оборот кодируется 13-ю разрядами. Они

установлены на место старых. Передаточные числа между осями телескопа и датчиками (см. схемы ниже) $240 \cdot \frac{72}{4} \cdot \frac{4}{24} \cdot \frac{125}{90} = 1000$ для часового угла и $240 \cdot \frac{125}{30} = 1000$ для склонений. Т.е.

датчики делают 1000 оборотов на оборот оси телескопа. Соответственно дискрет датчиков:

$$\frac{360 \times 3600}{2^{13} \times 10^3} = 0,158203125'' \approx 0,16'' .$$

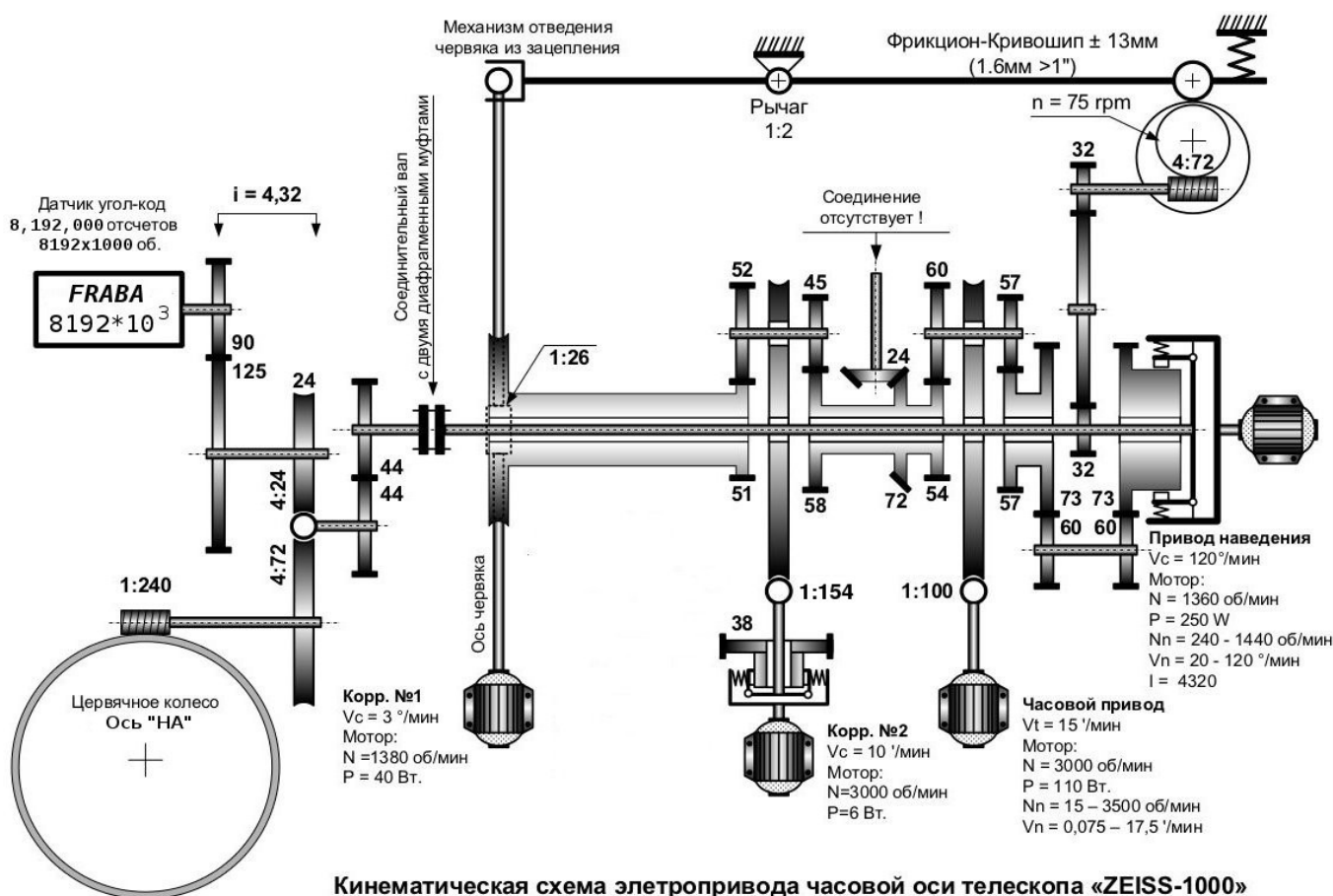
Поскольку датчики поставлены на место старых, т.е. через редукторы (см. схемы ниже), их реальная точность определяется качеством этих редукторов. После запуска новой системы управления был проведен эксперимент с целью попытаться оценить «вклад» редукторов в реальную точность датчиков. Двигатели телескопа включались на малую постоянную скорость, положения по осям записывались в файл с периодичностью раз в секунду. Затем из этих данных

вычиталась линейная регрессия. Оставались периодические вариации скорости вращения датчиков. Для их привязки к конкретным осям редуктора вычислялся Фурье-спектр. Оказалось, что «вклад» редукторов примерно 0.6" по часовому углу и около 1" по склонению ([подробнее см. в отчете за 2013г.](#)).

В интерфейсе наблюдателя [ZeissGUI](#) реализована панель [Encdrs](#) для представления реальных и расчетных положений угловых датчиков

Управление двигателями телескопа

Часовая ось телескопа управляется 4-мя двигателями, а ось склонений — 3-мя. Один двигатель маршевый для быстрого наведения он сидит прямо на главной оси редуктора. Остальные - для коррекции и часового ведения. Их вращение складывается через сложный дифференциальный редуктор. Ниже приводятся кинематические схемы этих редукторов восстановленные С.Драбеком на основе чертежей и практических исследований.



По этим схемам были выведены (в заложены в ZeissSEWdrive) формулы передаточных отношений от двигателей к осям телескопа.

Передача от главных червяков на оси телескопа $\frac{360 \cdot 3600}{240} = 5400 \text{ } \tau \text{ rev}$

Передача от валов двигателей наведения $\frac{5400}{\frac{72}{4}} = \frac{5400}{18} = 300 \text{ } \tau \text{ rev}$ Диапазон скоростей

$240 \div 1440 \text{ rpm} \rightarrow 1200 \div 7200^\circ/\text{сек} = 20' \div 2^\circ/\text{сек}$

Передача для асинхронного двигателя Коррекция1 НА $\frac{5400}{\frac{26}{1} \cdot \frac{52}{51} \cdot \frac{58}{45} \cdot \frac{60}{54} \cdot \frac{72}{4}} = \frac{5400}{683.3638} \approx 7.9 \text{ } \tau \text{ rev}$

Диапазон скоростей $20 \div 3000 \text{rpm} \rightarrow 0.16 \div 23.28''/\text{сек}$

Исходно ставилась задача сопровождения быстро движущихся объектов (ИЗС) двигателями коррекции. Их диапазоны скоростей существенно пересекаются, что делает задачу решаемой. К сожалению двигатель наведения тут использовать нельзя т.к. его минимальная скорость значительно выше максимальной скорости коррекции и кроме того его нужно останавливать чтобы сработала механическая муфта подключения дифференциального редуктора.

Управление такой двигательной установкой для решения этих задач представляло серьезную проблему. Разработан алгоритм одновременной работы 2-х двигателей коррекции на осях, позволяющий плавно менять скорость в полном диапазоне. Кроме того он и несколько расширяет этот диапазон скоростей. На часовой оси при малых изменениях скорости используется синхронный двигатель часового ведения т.к. он самый точный. Эти алгоритмы реализованы в классах организующих движение по осям *ZeissAxisHA* и *ZeissAxisDecl*.

Note: в специальном режиме «сервисный инженер» (имя пользователя *meh*) эти алгоритмы не используются, двигатели коррекции включаются отдельно — каждый в своем диапазоне скоростей.

Каждый SEW-привод управляется своим *Java*-объектом типа *ZeissSEWdrive*. Все они работают параллельно и реализуют обмен по CAN-шине по протоколу фирмы SEW каждый со своим приводом.

В интерфейсе наблюдателя реализована [панель Drives для представления состояния приводов](#).

Управление куполом

Управление приводом купола реализовано в классе *ZeissDome*. Оно происходит через модифицированный вариант класса *ZeissSEWdrive*. В который добавлено считывание кода АЦП аналогового входа AI1 SEW-контроллера. К этому входу подключен однооборотный аналоговый датчик угла. Он установлен на двигатель через редуктор 1:70. К сожалению реальная редукция на купол оказалась $12:834 = 1:69.5$. Это приводит к смещению нуля-пункта датчика на $\sim 2.6^\circ$ при полном обороте купола. Программа ведет и сохраняет в файле *Dome_shift.dgr* текущее накопленное смещение нуля-пункта. Изменение нуля-пункта производится при резком изменении напряжения на датчике с нуля до максимального (сейчас 9.17V) или обратно. Дискрет датчика $\approx 0.4^\circ$.

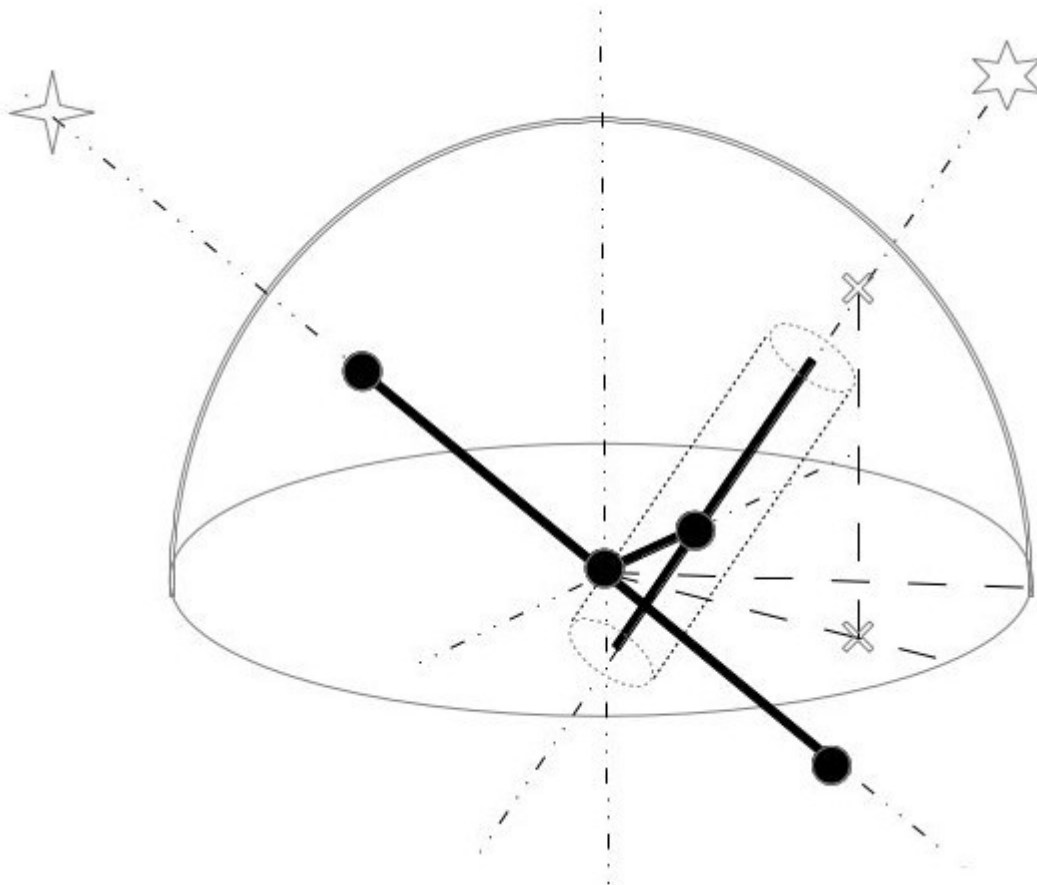
В программе реализовано несколько режимов управления куполом.

1. Программное управление заблокировано для используются кнопок аппаратного пульта (в подкупольном пространстве).
2. Ручное управление движением купола.
3. Установка купола в фиксированное положение по заданному азимуту.
4. Согласование купола с положением телескопа (купол повторяет движения трубы).
5. Согласование с координатами наблюдаемого объекта.

Последние два режима отличаются только источником координат HA,Dec с которыми нужно согласовать купол.

Из-за нецентрального положения трубы телескопа в подкупольном пространстве, согласование положения купола с телескопом не является тривиальной задачей. Азимуты телескопа и купола могут отличаться на любую величину от 0 до 180° (например при прохождении околосенитной области).

Разработан алгоритм, реализованный в классе *ZeissDome*, который непрерывно решает задачу из



аналитической геометрии о нахождении точки пересечения оси трубы телескопа со сферой купола. Проекция этой точки на горизонтальную плоскость дает азимут необходимого положения купола. Таким образом выполняется преобразование НА,Дес телескопа → азимут купола.

Чтобы не слишком часто «дергать» привод, программа включает движение если рассогласование между необходимым и измеренным положением достигает $\sim 1.6^\circ$ (4 дискрета датчика). Такая точность согласования трубы телескопа с забралом купола считается достаточной.

Скорость движения купола постоянная $\sim 0.85^\circ/\text{сек}$ (1100rpm на двигателе). Движение выключается при входе в зону $\sim \pm 0.8^\circ$ (2 дискрета датчика).

В интерфейсе наблюдателя [ZeissGUI](#) реализована панель **Dome** для управления куполом.

Коррекция наведения (Pointing Model)

Направления осей вращения телескопов всегда имеют статические ошибки и, кроме того, могут меняться во время работы из-за упругих гнущих конструкций. Поэтому для точного наведения на объект система управления должна содержать формулы коррекции отражающие специфику конструкции конкретного телескопа. Например в системе управления БТА это называется СКН (Система Коррекции Наведения). Коэффициенты в этих формулах отражают текущее состояние инструмента. Они определяются по результатам технических наблюдений звезд с точными координатами.

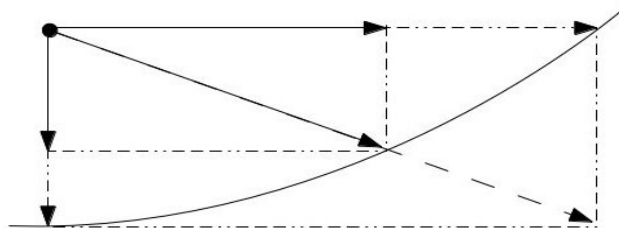
По результатам технических наблюдений и их обработки, были разработаны формулы коррекции наведения телескопа Цейсс-1000 (подробнее об этом [см. в отчете за 2012-й год](#)). Они основаны на классических формулах для экваториальных монтировок (см. например: <http://www.tpsoft.demon.co.uk/pointing.htm>) плюс гармоники за эксцентриситет и эллиптичность главных червячных колес.

Подробнее о формулах и коэффициентах можно прочесть в описании [панели РМ интерфейса ZeissGUI](#).

Ограничения движения

Для альт-азимутальных монтировок современных телескопов ограничения на повороты осей имеют тривиальный характер — просто аппаратные и/или программные концевики по координатам А и Z. Можно сказать что доступная область имеет прямоугольную форму в координатах А и Z. Управление скоростью может быть независимым по каждой координате.

Для экваториальных монтировок все значительно сложнее. Доступная область в координатах НА и Dec ограничена кривой линией. Аппаратные концевики невозможны (хотя на Цейсс-1000 такие попытки предпринимались). Остаются программные алгоритмы в системе управления. При этом должны учитываться как обычный режим работы, так и с перекладкой телескопа.



В программе должна быть определена линия ограничивающая доступную область. Программа должна следить за расстоянием до криволинейной границы в направлении вектора скорости, чтобы плавно остановиться и не переехать через эту границу. Если все таки телескоп оказался на границе или за ней, программа должна блокировать все направления движения кроме позволяющих вернуться в доступную область.

Ограничения телескопа (доступная область)

Область в которой можно наблюдать на телескопе Цейсс-1000 ограничена следующими причинами:

- с востока — куполом, точнее нижней кромкой проема забрала, т.к. труба телескопа опускается ниже ее;
- с запада — реальным горизонтом (труба тут поднимается выше кромки);
- с севера — опорной колонной полярной оси.

Таблица ограничений по всем этим причинам записана в в файле `/usr/local/ztcS/ZeissHorizon.tab`. В первом столбце «Decl» склонения от -46 до 225 через один градус. Значения <90 — область без перекладки, >90 — с перекладкой. Для каждого склонения в следующих двух столбцах «EastNA» и «WestNA» записаны ограничения по доступным часовым углам.

Таблица загружается в память сервера при старте. При этом сервер записывает в файл `restrDecl.tab` обратную таблицу: «HourAng» → «Decl.Bot.» «Decl.Top».

Note: ограничение в направлении на юг сейчас $Dec > -36^\circ$, но в специальном режиме «сервисный инженер» (имя пользователя *meh*) можно опускать трубу вплоть до горизонтального положения.

Графическое представление доступной области можно видеть на [панели Encdrs интерфейса ZeissGUI](#).

Для представления линии горизонта в программах [zeiss list](#) и [XEphem](#), файл `/usr/local/ztcS/ZeissHorizon.tab` был переработан в два файла: `zeiss_horizon_dir.hzn` — для нормальной моды наблюдений и `zeiss_horizon_rev.hzn` — для работы с перекладкой.

Ограничения навесной аппаратуры (запрещенные области)

На телескопе Цейсс-1000 аппаратура регистрации в фокусе Кассегрен подвешивается сзади трубы и увеличивает ее габарит. Кроме того кабели традиционно не подвешиваются к трубе, а свисают вниз и просто лежат на полу. Поэтому всегда существовала опасность зацепиться выступающими частями аппаратуры или порвать кабели.

Для того чтобы уменьшить вероятность таких неприятностей, был разработан, и реализован в системе управления, алгоритм запрещенных областей. Область должна задаваться замкнутой выпуклой ломаной линией в координатах HA,Dec. Алгоритм не должен позволять заезжать внутрь запрещенных областей. Сами области должны определяться на практике лицами ответственными за эксплуатацию телескопа.

Описания запрещенных областей находятся в файлах конфигурации методов наблюдений с именами типа: *ПРИБОР+ПРИЕМНИК+МАТОБЕСПЕЧЕНИЕ.conf* в справочнике */usr/local/ztcS/*. Количество областей может быть от 0 до 10. Имя параметра задающего область должно начинаться с «*region_*». Каждая точка это координаты через запятую. Точки линии разделяются «;». Т.е. строка описания выглядит как «*region_XXXXX=HA₁,Dec₁;HA₂,Dec₂;...;HA_n,Dec_n*» где HA_i, в часах, а Dec_i в градусах. Все значения это показания энкодеров телескопа. Последняя точка будет соединена с первой для получения замкнутой линии.

Например в файле */usr/local/ztcS/Photometer+CCD+DinaSystem.conf* для фотометра с ПЗС-матрицей записано:

```
region_BackEnd=22.33, -22.5;23.2, -22.5;23.2, -45.0;22.33, -45.0
region_Reverse=12.8,225.0;12.8,190.0;14.0,180;17.9,150.;17.9,255.
```

Первая область это ограничение для самой ПЗС-матрицы: нельзя между часовыми углами 22.33h и 23.2h опускаться ниже -22.5° по склонению. Вторая это ограничение по длине кабеля при работе с переключкой.

Административная подсистема

Для нормальной организации наблюдений на телескопе и предоставления корректных метаданных системам регистрации без излишней нагрузки на наблюдателя, в систему управления включена [подсистема администрирования наблюдениями](#). Она реализована в классе административного суб-сервера *ZeissAdminServer*. [Административная информация](#) включает списки наблюдателей, программ наблюдений, авторов программ, вариантов аппаратуры регистрации, а также расписание наблюдений. Последнее это список сетов наблюдений. Описание каждого сета это: дата начала сета, используемая аппаратура, список возможных наблюдателей, список программ наблюдений и их авторов.

В памяти сервера эта информация существует в виде структуры DOM. Она загружается в память системы при старте из XML-файла */usr/local/ztcS/ZeissAdmin.xml*. Ее редактирование выполняется через [Web-интерфейс](#) пользователем с уровнем доступа 5 (администратором телескопа).

[Панель Admin в интерфейсе наблюдателя ZeissGUI](#) позволяет использовать административную информацию во время наблюдений. Для этого в системе есть понятие «текущий сет» (*Current Set*). Информация из него идет и в прототип FITS-шапки, которая доступна в локальной сети на Windows-ресурсе *\\ztcS\ZEISS\FITS.HDR*, для включения в FITS-файлы системы регистрации.

Разработка клиентских приложений

Клиентские приложения, разработанные в 2010-2014 годах, написаны на нескольких языках программирования.

Java

[Основной интерфейс для наблюдателей и инженеров ZeissGUI](#) написан с использованием того же пакета реализации протокола XML-RPC на Java (*ws/xmlrpc*) что и сервер. Это графический клиент взаимодействующий по локальной сети с сервером системы управления Цейсс-1000. Он позволяет получать основную информацию о состоянии системы и выполнять управление работой телескопа во время наблюдений или сервисного обслуживания. Теоретически он может запускаться на любой машине в локальной сети под разными ОС, если в них установлена исполняемая система Java (JVM). И даже предусмотрена возможность в будущем запускать его как *Java-applet* в Web-браузере.

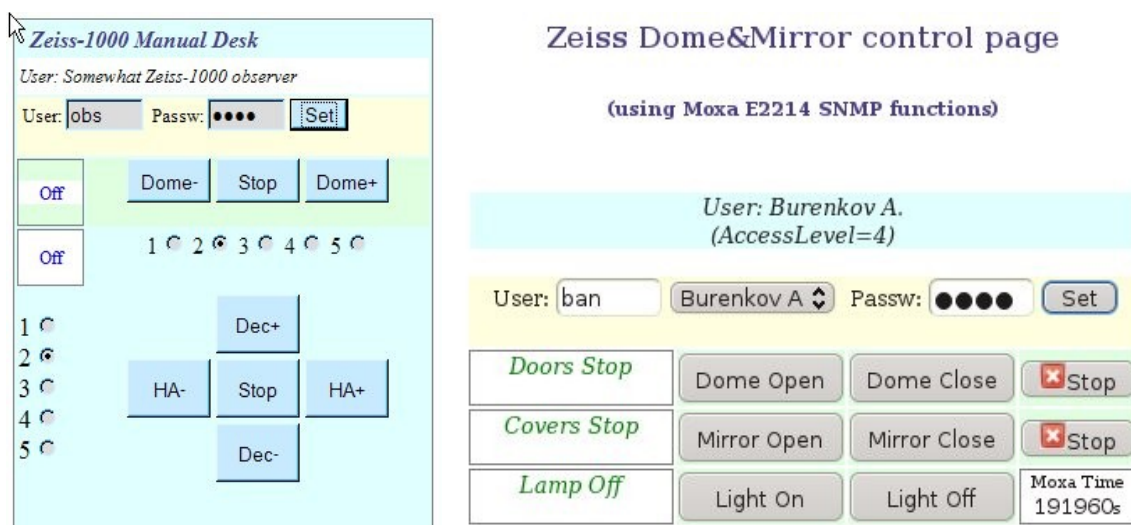
Клиент *ZeissOld* имитирует внешний интерфейс старой программы управления (через файлы на Windows-ресурсе \\ztcs\ZEISS\) и таким образом обеспечивает работу старых программ: *DinaSystem* - сбор с матрицы 2K на фотометре и UAGS; *tvguide* — гидирование по подсмотру UAGS; Web-интерфейс 20см ТВ-гида.

Клиент *ZeissFITSheder* непрерывно обновляет прототип FITS-шапки на Windows-ресурсе \\ztcs\ZEISS\ZFITS.HDR для использования в будущих системах сбора , прототип содержит всю административную, телескопную и метео информацию в корректном FITS-стандарте, собственно остается только добавить параметры системы сбора.

JavaScript

Доработан сторонний модуль *rpc.js* с использованием *Ajax* (XMLHttpRequest). С его помощью разработаны Web-интерфейсы:

- Web-приложение инженерного пульта которое может работать на мобильных гаджетах с WiFi под куполом телескопа. Приложение проверено на основных браузерах в *Android*, *iOS*, *Linux*, *Windows*.



- Web-интерфейс для удаленного управления забралом купола и крышками зеркала.
- [Web-приложение администратора на Цейсс-1000](#), которое предназначено для редактирования и просмотра административной информации в системе управления

телескопа. Административная информация включает списки наблюдателей, программ наблюдений, авторов программ, вариантов аппаратуры регистрации, а также расписание наблюдений.

Note: эти Web-приложения вызываются с [сайта компьютера системы управления](#), а XML-RPC сервер к которому они обращаются по Ajax имеет отличающийся URL. Современные браузеры, по соображениям безопасности, блокируют подобные «крос-серверные» обращения из Javascript. Необходимо чтобы в сервере было реализовано дополнение HTTP-протокола CORS (Cross-Origin Resource Sharing). К сожалению в используемом Apache XML-RPC (ws/xmlrpc) эта опция отсутствует. Java-класс `org.apache.xmlrpc.webserver.Connection` был доработан с включением поддержки CORS. Пока он загружается поверх основной библиотеки `ws/xmlrpc`, подменяя собой библиотечный.

Python

В основном пакете Python имеется штатный модуль `xmlrpclib`, при помощи которого написано несколько полезных скриптов.

- Разработан скрипт `ztcs_tycho2.py` облегчающий технические наблюдения. Он по заданному положению на небе (HA/Decl) находит в каталоге Tycho-2 ближайшую на данный момент звезду и запускает наведение на нее.
- Другой скрипт `ztcs_diff.py` применялся для исследования редукторов осей. Двигатели телескопа включались на малую постоянную скорость, положения по осям записывались в файл с периодичностью раз в секунду. Затем из этих данных вычиталась линейная регрессия. Оставались периодические вариации скорости вращения датчиков к которым применялся Фурье-анализ. Линии в полученных «спектрах» отождествлялись с периодами вращения конкретных валов в редукторах.
- Для оперативного наведения телескопа по координатам GRB-событий, [разработано клиентское программное обеспечение](#) на языке Python для работы с системой серверов GCN/TAN (GRB Coordinates Network / Transient Astronomy Network) [NASA](#). Разработка выполнена с использованием пакета [pygcn](#).

C

`zeiss1000xed`

Разработана на языке C с использованием библиотеки `xmlrpc_client` из пакета `xmlrpc-c`. Реализует FIFO-канал связи с телескопом для популярного эфемеридного пакета [XEphem](#).

Программа позволяет использовать почти все каталоги объектов формата `edb`. Кроме `edb`-формата «неподвижных» объектов, она воспринимает и формат каталогов объектов солнечной системы (кометы, астероиды) и околоземных (ИЗС). В этом случае она по заданным элементам орбиты рассчитывает не только текущие координаты, но и скорости смещения по RA и Dec. Все эти данные периодически пересчитываются и передаются в систему управления. Более подробное описание применения [XEphem](#) вместе с этой программой приведено [в этом документе](#).

C++

`zeiss1000stellarium`

Разработана на языке C++ с использованием библиотеки `xmlrpc_client++` из пакета `xmlrpc-c` и стандартной утилиты `xmlrpc_cpp_proxy` из пакета `xmlrpc-c-apps` которая

обращается к XML-RPC серверу, считывает служебную информацию о предоставляемых методах и создает прототипы C++ классов для разработки клиента. Программа реализует стандарт TCP-сервера связи с телескопами для компьютерного планетария *Stellarium*. Более подробное описание применения *Stellarium* вместе с этой программой приведено в этом документе.

zeiss_list

Также на C++, но графическая с использованием библиотеки Qt3. Реализует простое графическое представление списков объектов. Позволяет в два клика отправлять координаты объекта в систему управления. Подробное описание ее использования приведено в этом документе.

Shell

Можно писать обычные командные файлы с использованием стандартной утилиты *xmlrpc* из пакета *xmlrpc-c-apps*.

Служебные файлы

Все файлы имеющие отношение к серверу системы управления расположены на компьютере системы управления *ztcs* в справочнике */usr/local/ztcs/*.

ZeissServer.sh — командный скрипт для запуска или перезапуска сервера TCS Цейсс-1000 (используется в */etc/rc.d/rc.local*)

ZeissServerStop.sh — командный скрипт для ручного останова сервера.

ZeissTCS.conf — основной файл параметров конфигурации сервера. Его содержимое на текущий момент:

```
#Zeiss TCS configuration parameters
#Tue May 28 16:44:34 MSK 2013
iers.bulletinA=http\://maia.usno.navy.mil/ser7/ser7.dat
http.proxyHost=relay.sao.ru
http.proxyPort=8080
http.nonProxyHosts=*.sao.ru|localhost
zeiss.latitude=157157.3
zeiss.longitude=149195.4
zeiss.height=2055.0
zeiss.horizonFile=./ZeissHorizon.tab
zeiss.encodersType=CANopen
#zeiss.encodersType=Sim
zeiss.zeroCodeHA=129006
zeiss.zeroCodeDecl=1000641
zeiss.SEW7freq=0.9952898476
zeiss.parkHA=21\ :40\ :00.0
zeiss.parkDecl=+28\ :30\ :00.0
zeiss.Dome0=145.93
zeiss.parkDome=30.0
zeiss.pointingFile=ZeissPointing.conf
zeiss.usersFile=ZeissUsers.conf
zeiss.adminFile=ZeissAdmin.xml
zeiss.serviceUser=meh
```

Photometer+CCD+DinaSystem.conf — конфигурация метода «Фотометр с ПЗС матрицей»

```
#Zeiss Acquisition Hardware configuration
#Sat Dec 10 17:51:46 MSK 2011
```



```
fullname=Photometer+CCD+DinaSystem
name=Photometer+CCD
ident=CCD
soft=DinaSystem v2.2
corrHA=0.0
corrDecl=0.0
posAngle=72.5
parkHA=22\:00\:00.0
parkDecl=+22\:00\:00.0
parkDome=33.0
region_BackEnd=22.33, -22.5;23.2, -22.5;23.2, -45.0;22.33, -45.0
region_Reverse=12.8,225.0;12.8,190.0;14.0,180;17.9,150.;17.9,255.
```

UAGS+CCD+DinaSystem.conf — конфигурация метода «Спектрограф UAGS»(порядок строк параметров в файлах конфигурации не важен, он меняется при каждом спасении измененных параметров)

```
#Zeiss Acquisition Hardware configuration
#Wed Oct 08 00:12:41 GMT+04:00 2014
region_Reverse=18.001,90.2;17.999,90.2;17.999,225.0;18.001,225.0
name=UAGS+CCD
region_BackEnd=18.01,4.7;23.2,4.7;23.2, -45.0;18.01, -45.0
parkDecl=+39\:00\:00.0
ident=UAGS
corrDecl=-20.4
posAngle=0.0
parkHA=21\:40\:00.0
parkDome=33.0
soft=DinaSystem v2.2
corrHA=1.28
fullname=UAGS+CCD+DinaSystem
```

CEGS+CCD+DinaSystem.conf - конфигурация метода «Эшелле-спектрометр куда»

```
#Zeiss Acquisition Hardware configuration
#Sun Jun 08 23:38:23 GMT+04:00 2014
name=CEGS+CCD
parkDecl=+22\:00\:00.0
ident=CEGS1
corrDecl=-115.0
posAngle=0.0
parkHA=22\:00\:00.0
parkDome=33.0
soft=DinaSystem v2.2
corrHA=3.6
fullname=CEGS+CCD+DinaSystem
```

CEGS+PeltierCooledCCD+VLVSoft.conf — старый вариант метода «Эшелле-спектрометр куда»

ZeissAdmin.xml — XML-файл административных параметров

ZeissUsers.conf — список дополнительных (служебных) пользователей

```
#Zeiss main Users list (full observers list in Admin XML file)
#Thu May 26 15:18:16 MSD 2011
admin=*****|5|Zeiss-1000 main Administrator
vsher=*****|5|Shergin V.S.
meh=*****|4|Zeiss service engineer
obs=*****|3|Somewhat Zeiss-1000 observer
user=*****|2|Zeiss-1000 user (Set coord.only)
guest=guest|1|Guest user (Show only)
```

ZeissHorizon.tab — таблица ограничений области допустимых положений

ZeissPointing.conf — значения коэффициентов формул коррекции наведения

```
#Zeiss Pointing Model coefficients
#Calculated by zeiss_lsqm74 (WITHOUT! reverse mode account)
# IH, ..., DAF - only!
# GWxxx - preserved from previos (May 2012) version
#Thu Jul 25 22:01:28 GMT+04:00 2013
IH=-73.93
ID=31.19
CH=130.40
NP=1.25
MA=-12.47
ME=-66.09
TF=6.30
DAF=94.44
GWH1=31.70
GWH2=5.35
GWH2phi=3.8
GWD1=12.21
GWD1phi=65.0
GWD2=-4.45
GWD2phi=-20.0
```

Dome_shift.dgr — текущее накопленное смещение нуля-пункта положения купола.

ser7.dat — файл с данными Службы Вращения Земли (IERS), скачивается автоматически при старте сервера с [сайта IERS](#).

*lib*_jni.so* — JNI-библиотеки написанные на C и подключаемые к серверу на Java.

ZeissServer.log — файл протокола работы сервера управления.

ZeissGUI.sh — командный скрипт старта интерфейса наблюдателя (обычно вызывается через ссылку */usr/local/bin/zgui*).

fromhostname - командный скрипт определяющий с какого компьютера зашел пользователь (используется в *ZeissGUI.sh*).

zeiss1000xed, *zeiss1000stellarium*, *zeiss_list* — клиентские программы (см. выше), вызываются через ссылки в */usr/local/bin/*.

Загрузка

Перед стартом самого сервера системы должны быть загружены:

- can.ko* — драйвер CAN-карты в ядро системы
- zeiss_can_io_net* — даемон ввода/вывода по CAN-шине
- bta_control_net* — даемон сетевой связи с АСУ БТА

Все это реализовано в стартовом файле ОС Linux (*/etc/rc.d/rc.local*), который в настоящий момент выглядит так:

```
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

# Check: if no CAN-driver => load it (with parametes from /etc/modprobe.conf)
```

```
[ -d /sys/module/can ] || modprobe can
sleep 1
# If CAN Ok start Zeiss CAN I/O daemon
[ -d /sys/module/can ] && cd /usr/local/sbin && ./zeiss_can_io_net
>/var/log/zeiss_can_io.log 2>&1 &

# Starting BTA TCS network daemon
/usr/local/sbin/bta_control_net acs7.sao.ru >/var/log/bta_control_net.log 2>&1 &

# Starting Zeiss-1000 TCS Server
(cd /usr/local/ztc; su vsher -c "./ZeissServer.sh simulator" >/dev/null 2>&1 ) &

# Starting meteo-station daemon
/usr/local/bin/wviewd_watching.sh >/dev/null 2>&1 &

chmod a+r /dev/audio /dev/adsp /dev/dsp /dev/mixer
chmod a+w /dev/mixer

touch /var/lock/subsys/local
```

Также должен работать сервис `wview` [метеостанции Vantage Pro](#).

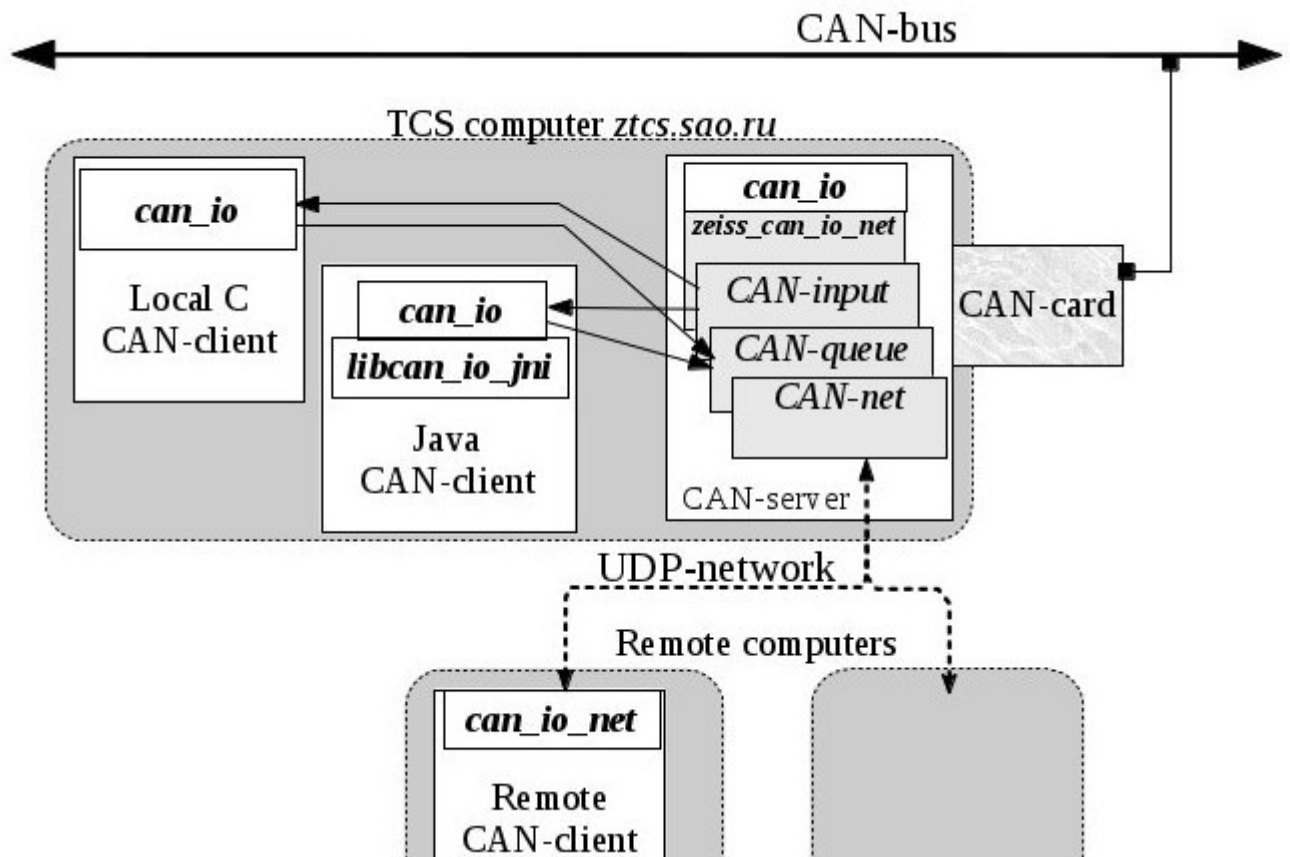
МО CAN-шины

Для доступа к устройствам на CAN-шине в управляющем компьютере `ztc` установлена карта *Adlink PCI-7841*.

Для работы с этой картой в ядро системы установлен сторонний драйвер [LinCAN](#) (автор [Pavel Pisa](#) <pisa@cmp.felk.cvut.cz>).

Драйвер допускает использование только одной программой. Для одновременной параллельной работы нескольких программ с устройствами на CAN-шине разработаны библиотека `can_io` и CAN-сервер `zeiss_can_io_net`.

Модуль `can_io` включается (компонуется) в программы написанные на языке C. В нем реализованы интерфейсы обмена CAN-фреймами с между CAN-клиентами и CAN-сервером. Для клиентов на *Java* модуль `can_io` включен в JNI-библиотеку `libcan_io_jni` и написан *Java*-класс `CANio` для ее подключения. Демон `zeiss_can_io_net` запускается при старте системы. Он порождает (*fork*) процессы-копии для получения (*input*) и отправки (*queue*) CAN-фреймов.



CAN-сервер также может организовывать и сетевой обмен CAN-фреймами в виде UDP-пакетов. Для этого порождается еще один процесс взаимодействующий по сети со специальным вариантом клиентской библиотеки *can_io_net*. Сейчас эта возможность реально не используется.