Sparkle: консольная утилита для управления EMCCD детектором Andor iXon+897

Сафонов Б.С.

18 января 2014 г.

1 Введение: актуальность и требования

Программа Sparkle предназначена для получения изображений с детектора Andor iXon+897 в ходе экспериментов, связанных с разработкой Многорежимной Быстрой Камеры, и является чисто инженерным, временным приспособлением. Она является в некоторым смысле последователем другой инженерной программы, которую будем называть условно generic2. generic2 — это несколько улучшенная версия примера управляющей программы, поставлявшейся вместе с SDK (улучшение заключалось в добавлении кое-какого функционала).

generic2 дает максимальную свободу управления детектором, она фактически является лишь прослойкой, позволяющей из командной строки вызывать функции драйвера, перечисленные в [1]. В тоже время опыт наблюдений показал, что применение ее в эксперименте, не связанном с детектором непосредственно, неудобно, т.к. установка параметров детектора размыта по многим командам. Кроме того, это небезопасно, т.к. легко получить некорректные данные или ввести детектор в режим из которого он выводится только перезагрузкой.

В связи с этим возникла потребность в программе, лучше отвечающей нуждам физического и астрономического эксперимента. Перечислим основные требования к ее устройству.

1. Установка всех параметров детектора, необходимых для получения изображений, должна осуществляться минимальным количеством действий пользователя.
2. Должна быть возможность прописать параметры детектора в конфигурационном файле.
3. Требуется предусмотреть возможность изменять большое количество параметров одной командой.
4. Должна оставаться возможность изменять отдельные параметры детектора.

Первые два требования обеспечивают *однородность* обстоятельств измерений (под обстоятельствами измерений/экспериментов мы понимаем набор параметров детектора). Пользователь может использовать один и тот же конфигурационный файл в нескольких экспериментах и быть уверенным, что эти эксперименты выполнены при одинаковых обстоятельствах. Также первые три требования значительно уменьшают объем взаимодействия пользователя с программой, не отнимая время и внимание, необходимые для проведения собственно эксперимента. Четвертое требование оставляет свободу изменения отдельных параметров детектора.

Эти требования удовлетворены при разработке Sparkle, описанию которой и посвящен данный документ. В разделе 2 приведено описание общего устройства программы. В разделе 3 описан функционал программы и пользовательский интерфейс.

2 Устройство программы

Программа написана на языке C++, что обусловленно тем, что SDK, поставляющееся с детектором, представляет собой библиотеку для C и generic2, упоминавшаяся ранее, также написана на C. C++ позволяет использовать это “наследие” и в тоже время писать более структурированный код. Для упрощения работы со строками применена стандартная библиотека STL.

Таблица 1: Команды управления режимами.

команда	описание
`rlist`	Перечислить имена имеющихся режимов.
`rnew name`	Создать новый режим с именем `name`.
`rdel name`	Удалить режим `name`.
`rmod name`	Сделать режим `name` текущим.
`rcopy name`	Создать копию текущего режима с именем `name`.
`rload path`	Загрузить режим из файла `path`, путь абсолютный. О формате конфигурационного файла см. раздел 3
`rprint <name>`	Вывести все параметры режима `name`. Если `name` не указано, параметры текущего режима.
`rval <name>`	Произвести валидацию режима `name`. Если `name` не указано, использовать текущий режим.
`rapp <name>`	Загрузить режим `name` в детектор. Если `name` не указано, использовать текущий режим.

Пользовательский интерфейс реализован в виде командной строки, поскольку 1) это значительно упрощает разработку, 2) вследствие ограниченности функционала удобство использования не страдает от этого. Соответственно, базовая структура программы — это цикл, по очереди обрабатывающий команды, поступающие от пользователя.

Основным понятием в Sparkle является **режим** — совокупность параметров, полностью конфигурирующая детектор для получения изображения. В программе существует контейнер, содержащий произвольное количество режимов у каждого из которых есть уникальный идентификатор — имя. Кроме того, есть текущий режим, доступный в данный момент для редактирования. Имя текущего режима отображается перед символом `<`, который, в свою очередь, располагается перед командами, вводимыми пользователем (см. конец раздела 3). Пользователь может совершать действия с режимами, подавая соответствующие команды, см. табл. 1.

В программе предусмотрена возможность редактирования текущего режима путем подачи команд изменения параметров, они перечислены в разделе 3. Проверка заданных параметров производится одновременно, с помощью отдельной команды `rval`. Это сделано в связи с тем, что параметры сильно связаны между собой и корректность может быть установлена только для их полной совокупности.

Загрузка режима в детектор выполняется командой `rapp` (в начале выполнения этой команды автоматически производится валидация). После успешной загрузки детектор готов к получению изображений, что может быть осуществлено двумя с помощью одной из соответствующих команд, см. раздел 3. При запуске программы в контейнер кладется режим с именем default, который в этот момент является текущим. Этот режим удалить нельзя.

Как видно, концепция режимов позволяет естественным образом удовлетворить первым трем требованиям из раздела 1. Четвертое требование удовлетворяется возможностью редактировать режим из программы.

Вследствие того, что детектор является прибором, работающим в реальном, а значит, принципиально несовершенном мире у него есть некоторые существенные особенности, которые несколько нарушают логику системы режимов, они приведены в следующем разделе, посвященном собственно функционалу программы.

3 Функционал

В таблице 2 приведены команды установки параметров режима, соответствующий диапазон допустимых значений и описание. Описание команды также будет выведено программой если второе слово в введенном тексте отсутствует или не является числом. Обсудим отдельно установку температуры.

Температура относится к параметрам детектора, которые не могут быть установленны мгновенно. Поэтому в том случае если температура режима отличается от текущей температуры детектора или температура детектора не стабилизирована при загрузке режима (команда `rapp`) будет предпринята попытка стабилизировать температуру на уровне, заданном соответствующим параметром. В процес-

Таблица 2: Команды установки параметров режима.

команда	тип	диапазон	описание
`fitsname name`	строка		`name` — имя FITS файла, в который будет производиться запись изображений при выполнении команды `acq`. Расширение *.fits добавляется автоматически. Имя может содержать буквы, цифры, подчеркивания.
`rtaname name`	строка		`name` — имя FITS файла, в который будет производиться запись изображений при выполнении команды `prta`. Расширение *.fits добавляется автоматически. Имя может содержать буквы, цифры, подчеркивания.
`fitsdir path`	строка		`path` — абсолютный путь к директории, в которую будут записываться файлы при выполнении команд `acq` и `prta`. Директория должна существовать.
`rtaSkip val`	целое	1-1000	При получении изображений по команде `prta` записывается только кадры, номер которы кратен `val`.
`numKin val`	целое	1-16000	`val` — количество кадров в серии, при выполнении команды `acq`.
`shutter val`	целое	0, 1	0 — затвор закрыт, 1 — затвор открыт.
`ft val`	целое	0, 1	Режим полнокадрового переноса: 0 — выключен, 1 — включен.
`ampl val`	целое	0, 1	Детектор имеет два АЦП, один — с регистром электронного усиления (`val` = 0), другой — без (`val` = 1).
`horSpeed val`	целое	0, 1, 2	Скорость горизонтального переноса, определяет скорость считывания. При `ampl` = 1 доступна только `val` = 0, что соответствует 3 MHz, при tt ampl = 0: `val` = 0 — 10 MHz, 1 — 5 MHz, 2 — 3 MHz.
`preamp val`	целое	0, 1, 2	Предусиление, величины см. в reference sheet [2].
`vertSpeed val`	целое	0, 1	Скорость вертикального переноса: `val` = 0 — 0.3 мкс/строку, `val` = 1 — 0.5 мкс/строку, `val` = 2 — 0.9 мкс/строку, `val` = 3 — 1.7 мкс/строку, `val` = 3 — 3.3 мкс/строку,
`vertAmpl val`	целое	0-4	Напряжение вертикального переноса.
`EMgain val`	целое	1-1000	Электронное усиление.
`temp val`	целое	-80-0	температура детектора в градусах Цельсия.
`exp val`	double	0.001-1000.0	Экспозиция в секундах.
`imLeft val`	целое	1-512	Номер столбца, соответствующего левому краю считываемого изображения.
`imRight val`	целое	1-512	Номер столбца, соответствующего правому краю считываемого изображения.
`imBottom val`	целое	1-512	Номер строки, соответствующей нижнему краю считываемого изображения.
`imTop val`	целое	1-512	Номер строки, соответствующей верхнему краю считываемого изображения.

се установки температуры на экране будет обновляться ее текущее значение и состояние охладителя, командная строка при этом будет недоступна. Как только температура стабилизируется на заданном уровне, режим будет считаться успешно установленным, и управление будет возвращено пользователю. Процедуру мониторинга установки температуры можно прервать нажатием любой клавиши, охлаждение/нагрев при этом будут остановлены, управление возвращено пользователю.

После успешной загрузки режима можно запустить получение изображений одной из команд: `prta` или `acq`. При попытке выполнить одну из этих команд для незагруженного детектора будет выдана ошибка.

Команда prta запускает детектор в моде run till abort, т.е получение изображений будет вестись непрерывно пока пользователь не прервет его. На диске при этом сохраняется последнее полученное изображение, причем только если его номер в серии кратен параметру `rtaSkip`. Сохранение производится в файл, указанный командой `rtaname`, в директорию, указанную командой `fitsdir`. Прервать получение изображений можно нажав любую клавишу. Команду prta удобно применять в тех случаях, когда нужно центрировать объект в поле зрения, сфокусировать телескоп и т.п., т.е. при подготовке к получению серии.

Команда acq запускает получение серии кадров длиной `numKin`, результат сохраняется в файл, указанный командой `fitsname`, в директорию, указанную командой `fitsdir`. Сохранение изображений на диск производится параллельно с получением изображений. Выполнение этой команды также можно прервать, нажав любую клавишу.

Фактически устанавливаемая экспозиция и период между кадрами могут отличаться от экспозиции, заданной командой `exp`, поскольку эти параметры не могут быть произвольными. Например, при включенном полнокадровом переносе экспозиция не может быть меньше времени, требующегося на считывание кадра. Существуют и другие ограничения, подробнее см. [1]. Поэтому для удобства использования добавлена команда `tim`, которая выводит на экран фактическую экспозицию (exposure) и период между кадрами (kinetic cycle time). Команда может быть вызвана только для успешно загруженного режима.

Завершается работа с помощью команды `exit`. Важной особенностью детектора является то, что он должен быть контролируемо нагрет до температуры окружающей среды перед выключением. Поэтому в момент введения команды `exit` текущая температура детектора t_c сравнивается с температурой в момент запуска программы t_b, их разницу обозначим Δt. Если $|\Delta t| > 15^\circ$ детектору подается команда нагреться до температуры $t_b - 10^\circ$. При этом неявно предполагается, что t_b является хорошей оценкой температуры окружающей среды, что впрочем может быть неверно, особенно в некоторых нестандартных ситуациях (множественные запуски программы, например). Однако другого способа оценить температуру окружающей среды у нас все равно нет.

Конфигурационный файл имеет следующий формат. Первая строка должна содержать единственное слово, обозначающее имя режима, который будет создан при загрузке этого файла. Все дальнейшие строки представляют собой команды установки параметров, перечисленные в таблице 2, по одной на строку. Допускаются комментарии, после знака %. Пример содержимого конфигурационного файла:

```
test % name of regime
fitsname test % name of FITS file
rtaname rta % etc
fitsdir /home/safonov/username/
rtaSkip 1
numKin 10
shutter 0
ft 1
ampl 1
horSpeed 0
preamp 0
vertSpeed 1
vertAmpl 3
temp -1
exp 0.1
EMgain 1
```

В завершение приведем приведем пример работы с программой:

```
default<rload /home/username/Sparkle/def.rgm
default<rlist
default
test
default<rmod test
test<rval
test<rapp
test<acq
test<exit
```

Список литературы

[1] Andor, “Andor user’s guide to software development kit,”

[2] Andor, “Andor iXon+897 X-5947 Performance sheet,”