

НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЕ УЧРЕЖДЕНИЕ
ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ им. Г.И.Будкера СО РАН

В.Р. Мамкин

ИСПОЛЬЗОВАНИЕ VME
И САМАС КОНТРОЛЛЕРОВ
В СОСТАВЕ EPICS

ИЯФ 2001-12

НОВОСИБИРСК
2002

Использование VME и CAMAC контроллеров в составе EPICS

В.Р. Мамкин

Институт ядерной физики им. Г.И. Будкера
630090 Новосибирск, Россия

Аннотация

В статье рассмотрена возможность применения контроллеров CAMAC и VME, разработанных в ИЯФ, в составе системы сбора данных EPICS. Приведены характеристики контроллеров. Описано системное программное обеспечение контроллеров и особенности его использования вместе с EPICS.

VME and CAMAC controllers employment in the EPICS

V.R. Mamkin

Institute of Nuclear Physics, 630090, Novosibirsk

Abstract

The possibility of VME and CAMAC controllers employment in the EPICS data acquisition system is considered. Controllers were developed in the BINP. Technical characteristics of controllers are presented. System software and it using with EPICS is described.

© *Институт ядерной физики
им. Г.И. Будкера СО РАН, Россия*

1 Введение

EPICS (Experimental Physics and Industrial Control System) является системой сбора данных и управления, широко применяемой в области научной автоматизации. EPICS представляет собой большой программный проект, хорошо приспособленный для построения распределенных систем управления. Взаимодействие между компонентами системы осуществляется по протоколам сети TCP/IP. Одним из основных элементов системы является контроллер ввода-вывода (IUC), который выполняет функции управления, а также выступает посредником между аппаратурой и сетью TCP/IP. Разработанные автором интеллектуальные контроллеры САМАС и VME, могут служить аппаратной основой для IUC. Низкая стоимость самих контроллеров, а также использование свободно распространяемого программного обеспечения (freeware) позволяют значительно снизить затраты в случае широкого внедрения EPICS в ИЯФ.

2 Архитектура EPICS

Подробно архитектура EPICS описана в [1]. Основными компонентами системы являются: интерфейс оператора (ОПИ), контроллер ввода-вывода (IUC) и сеть Ethernet с протоколами TCP/IP (рис. 1). В одной сети может присутствовать несколько интерфейсов оператора и контроллеров IUC. Назначение ОПИ состоит в хранении данных, их представлении и off-line обработке. В большинстве случаев ОПИ являются клиентами, которые запрашивают данные у IUC. Обычно интерфейсом оператора являются UNIX рабочие станции. Назначение IUC состоит в непосредственном сборе данных с аппаратуры, их on-line обработке и управлении. IUC, с одной стороны, обращен к аппаратной части установки, а с другой стороны, подключен к сети TCP/IP. IUC в большинстве случаев является сервером, который отвечает на запросы от ОПИ и других IUC. Контроллером IUC обычно является VME крейт под управлением ОС VxWorks.

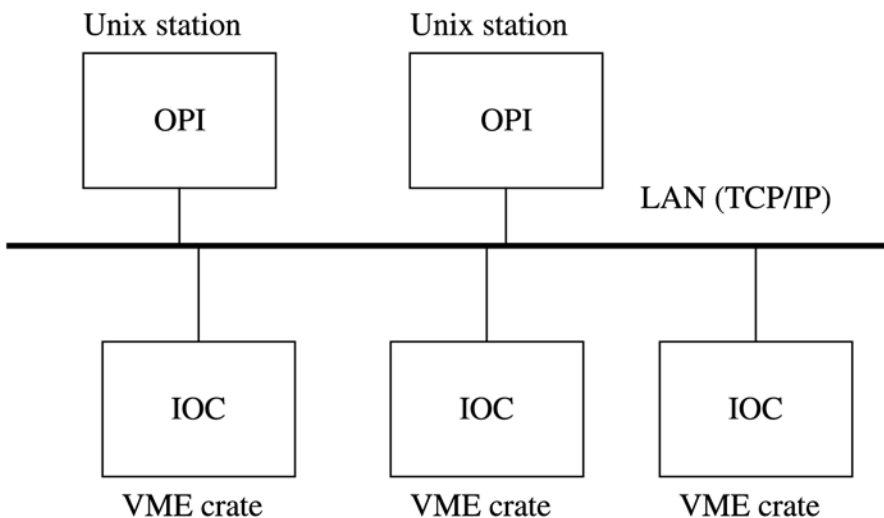


Рис. 1: Компоненты EPICS.

3 Контроллер ввода-вывода (IOC)

Ядром IOC является база данных (рис. 2). База данных содержит результаты измерений, а также задает алгоритмы обработки данных и управления. Элементом базы данных является запись. Запись содержит логически связанные данные и сопровождающий их программный код. Записи могут выполнять функции обработки данных, управления или вычислений. Внешний доступ к базе данных осуществляется через модуль Channel Access (CA). Внутренний доступ к базе данных может осуществляться драйверами устройств.

На нижнем уровне доступ к аппаратуре контролируется драйверами устройств, которые осуществляют непосредственный доступ к регистрам и памяти. Полученные от аппаратуры данные попадают в запись базы данных, которая соответствует данному драйверу устройства.

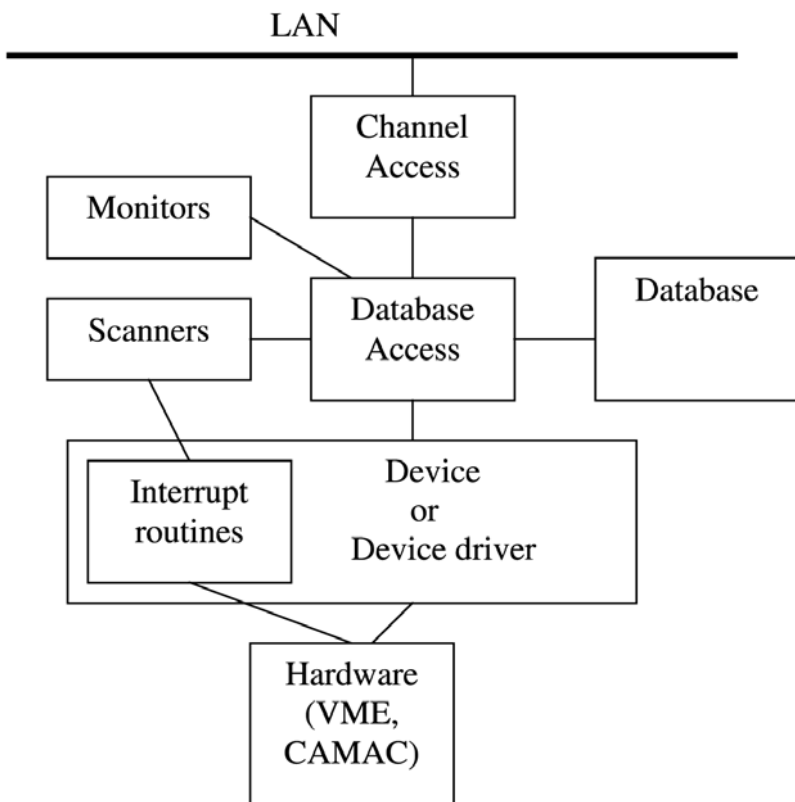


Рис. 2: Упрощенная структура ИОС.

Прикладной программист, использующий ИОС, имеет дело с разработкой следующих компонентов. При добавлении в контроллер аппаратных модулей разрабатываются соответствующие им драйверы устройств. Проектируется база данных, т.е. система записей, необходимых для обработки данных и управления. Структура записей помещается в файлах-описаниях базы данных (dbd файлы), которые загружаются в систему перед запуском ИОС. При проектировании базы данных можно использовать либо стандартный набор записей, который поддерживан в ИОС, либо можно разработать собственные типы записей. Примерами стандартных записей могут служить типы analog input, analog output, digital input, digital output, которые используются для ввода-вывода информации из

аппаратных модулей. При управлении сложной аппаратурой может понадобиться разработка собственных типов записей. В этом случае кроме файла-описания dbd, программист разрабатывает текст записи на языке С, в котором описываются операции по обработке данных.

С точки зрения стоимости системы, ИОС имеет наибольший вес, так как обычно контроллером ИОС является VME крейт с микрокомпьютером на базе процессора Motorola, под управлением ОС VxWorks. К стоимости VME аппаратуры здесь добавляется стоимость лицензии VxWorks. Кроме того, количество установленных ИОС пропорционально количеству каналов в системе сбора данных. В качестве ОПИ могут выступать обычные рабочие станции, под управлением ОС общего назначения. По этой причине основной интерес представляет возможность удешевления ИОС.

4 Аппаратная платформа для ИОС

4.1 VME контроллер

Основные характеристики контроллера приведены в таблице.

Процессор	MC68EN360 Тактовая частота 33Мгц
ОЗУ	8 (16) Mb
BootROM	128 Kb
Flash	8 Mb flash disk
ИО	Ethernet 10 base T 2 x RS232
other	RTC 2 kb NVRAM
VME	A16, A24, A32 D8, D16, D32 VME requester - ROR VME arbiter - PRI or round robin

Структура контроллера приведена на рисунке 3.

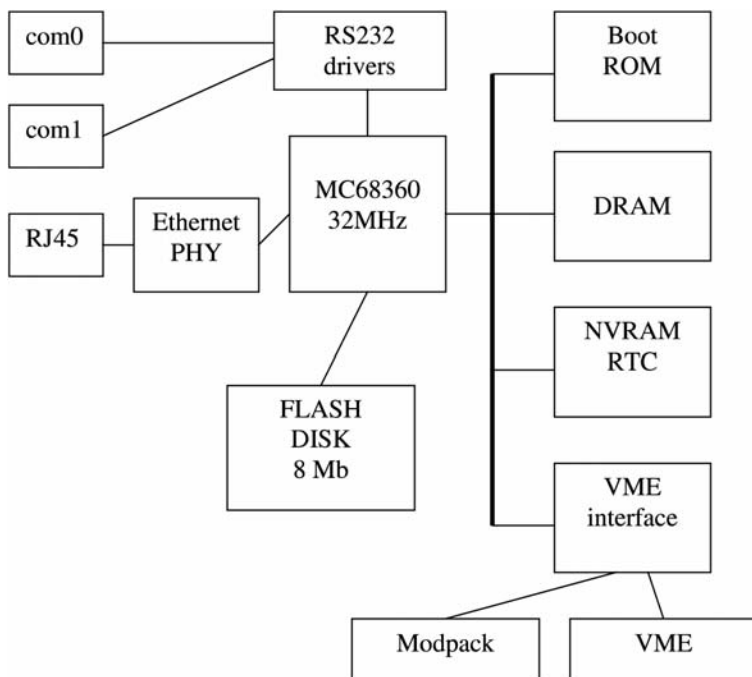


Рис. 3: Блок-схема контроллера VME.

4.2 CAMAC контроллер

Основные характеристики контроллера приведены в таблице.

Процессор	MCF5307 RISC, 66 MHz
ОЗУ	16 Mb
BootROM	512 Kb
Flash	4 Mb flash disk
IO	Ethernet 10/100 Console RS232
other	RTC 2 kb NVRAM
CAMAC	R24, W24 C, Z, I LAM

Структура контроллера приведена на рисунке 4.

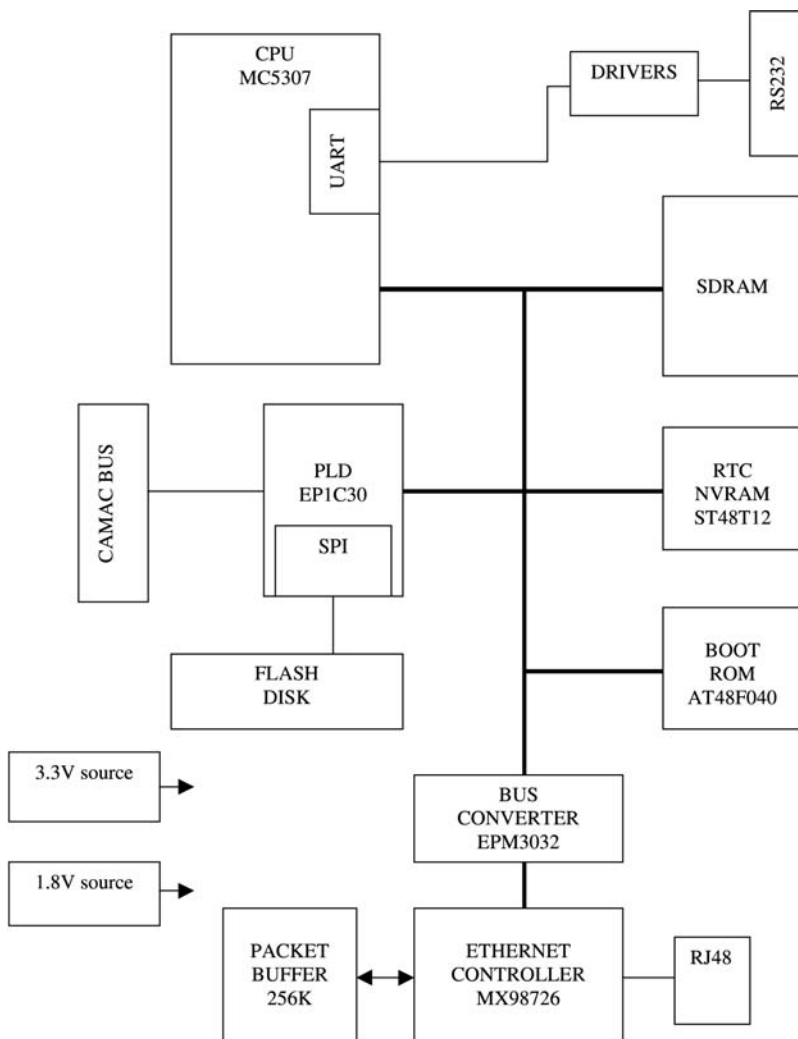


Рис. 4: Блок-схема контроллера CAMAC.

5 Системное программное обеспечение

ИОС базируется на операционной системе реального времени. До версии EPICS 3.14 операционной системой могла являться VxWorks. В версии 3.14 EPICS предприняты дополнительные усилия для увеличения переносимости ИОС на различные платформы. Например, все внутренние системные вызовы ИОС происходят через программный интерфейс, не зависящий от операционной системы (OSI уровень). OSI в свою очередь, обращается к уровню OSD, т.е. зависящему от ОС. При переносе EPICS на другую систему разработчик должен переписать заново только уровень OSD, оставив неизменными все остальные части ИОС. С точки зрения ИОС, операционная система должна обладать следующими свойствами: многозадачность, наличие бинарных семафоров, программно задаваемые интервалы, поддержка TCP/IP и интерфейса BSD сокетов.

5.1 RTEMS

Начиная с версии 3.14, в качестве операционной системы для ИОС может использоваться RTEMS. RTEMS (Real Time Executable for Multiprocessor Systems) представляет собой ОС реального времени, распространяемую с открытым кодом. RTEMS имеет следующие особенности:

- многозадачность с перепланировкой по событиям или таймеру
- распределение процессорного времени в соответствии с приоритетами задач
- межзадачное взаимодействие и синхронизация
- средства управления памятью
- поддержка драйверов внешних устройств
- гибкость конфигурации

При работе с RTEMS программист оперирует со следующими объектами операционной системы:

Задачи. Можно создавать, удалять, останавливать задачи, менять их приоритет.

Прерывания. ОС дает возможность устанавливать обработчики прерываний, запрещать или разрешать прерывания.

Часы реального времени. С использованием средств операционной системы прикладная задача может получать или устанавливать показания часов реального времени.

Таймеры. Для вызова через заданное время определенной функции используются таймеры. В отличие от аппаратных таймеров, количество программных таймеров не лимитировано и задается при конфигурации ОС.

Семафоры используются для организации доступа нескольких задач к разделяемому ресурсу или для синхронизации задач.

Очереди сообщений могут использоваться для передачи информации между задачами, а также для синхронизации задач.

События также могут использоваться для синхронизации задач, но, в отличие от семафоров, события могут инициироваться обработчиками прерываний, т.е. аппаратным уровнем.

Сигналы. Каждая задача может установить обработчик сигнала, а также послать сигнал. Сигналы служат для обработки асинхронных программных событий.

Отделы представляют собой целостную область памяти, разделенную на буферы одинакового размера. Задачи могут запрашивать у отдела буфер и возвращать его.

Регионы представляют собой целостную область памяти, разделенную на блоки, размер которых может варьироваться. Задачи могут запрашивать и освобождать блоки памяти. В отличие от стандартной C кучи, если в регионе не хватает памяти, задача может быть приостановлена до тех пор, пока память не появится.

Драйверы устройств предназначены для абстрагирования прикладной задачи от аппаратуры. Задача может открывать, закрывать, писать и читать драйвер устройства.

Кроме того, операционная система предоставляет интерфейс сокетов TCP/IP. Подключение сетевых сервисов происходит путем компоновки объектного кода прикладной программы и сетевой библиотеки. Программный интерфейс сокетов не отличается от принятого в BSD UNIX.

5.2 Средства разработки приложений

Для RTEMS используются свободно распространяемые GNU компилятор, линкер, ассемблер и ряд других утилит. Особенность этого ПО состоит в том, что оно построено по идеологии host-target. Платформа, на которой выполняется компилятор является host платформой. Для RTEMS имеются средства разработки для host Windows и Linux x86. Целевой (target) системой является система, для которой генерируется код, т.е. RTEMS- motorola.

6 Особенности ИОС на базе RTEMS

6.1 Создание нового проекта

Новый программный проект с использованием EPICS под управлением RTEMS должен содержать следующие компоненты: системные библиотеки RTEMS, библиотеки ИОС, dbd файлы с описанием новых типов записей, исходные С файлы записей, исходные С файлы устройств (см. рис. 5). Эти компоненты собираются линкером в единый исполняемый файл, который далее загружается в контроллер.

Ядро ИОС собрано в несколько библиотек:

- libCom
- libasIoc
- libca
- libcas
- libdbIoc
- libdbStaticIoc
- libdbtoolsIoc
- libgdd
- libiocsh
- libmiscIoc
- librecIoc
- libregistryIoc
- libsrvIoc
- libsoftDevIoc
- libtestDevIoc
- librtemsCom

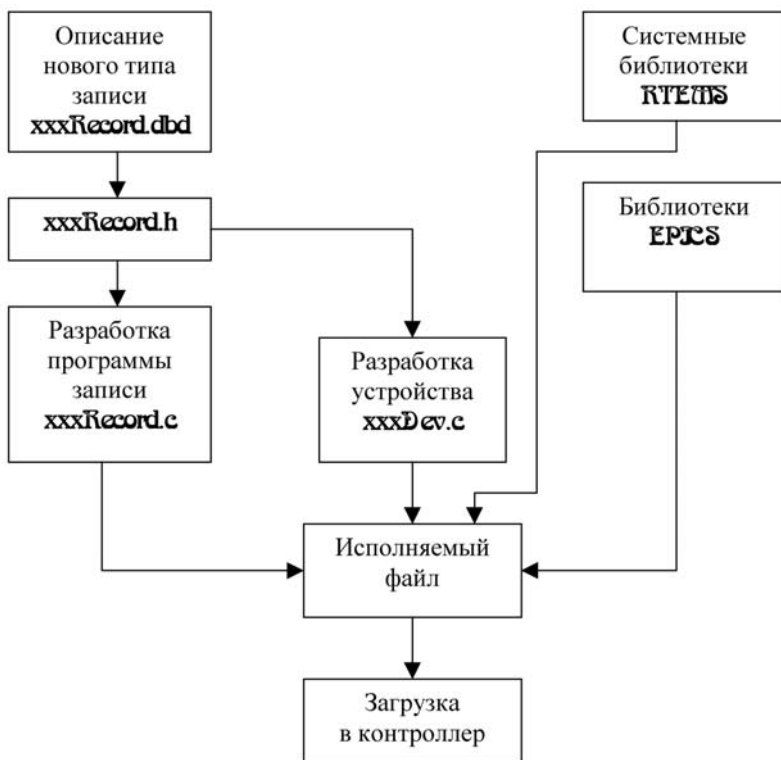


Рис. 5: Создание нового проекта.

Среди этих библиотек `librtemsCom` и `libiocsh` специфичны для ОС RTEMS. Библиотека `librtemsCom` содержит код инициализации и параметры конфигурации RTEMS (максимальное количество файлов, семафоров, задач и т.д.). Библиотека `libiocsh` содержит специально разработанный для RTEMS командный интерпретатор (shell). В ОС VxWorks при работе с ИОС используется стандартный shell, который имеет возможность интерпретировать синтаксис языка C и непосредственно из командной строки вызывать библиотечные функции. Поскольку в RTEMS такой возможности нет, для ИОС был разработан специальный shell, который выполняет различные сервисные команды и позволяет отслеживать текущее состояние ИОС.

В программный проект могут входить также разработанные пользователем типы записей и драйверы устройств.

Разработка нового типа записи начинается с его описания в файле dbd, например, xxxRecord.dbd. Синтаксис dbd файла приведен в документации на базу данных EPICS (см.[2]). После создания описания записи, с помощью утилиты dbToRecordtypeH из dbd файла получают h-файл, xxxRecord.h. Этот файл содержит описания структур и типов записи записи в правилах языка C. Далее файл xxxRecord.h должен использоваться в всех C файлах, где происходит обращение к полям записи.

Кроме dbd файла, где описывается структура новой записи, должен быть создан исходный текст записи на языке C. В документации EPICS (см.[1]) приведены примеры разработки исходных текстов записей. При разработке используются структуры данных, описанные в автоматически созданном файле xxxRecord.h.

После создания записей, прикладным программистом должны быть разработаны драйверы устройств или так называемые устройства – программные модули, которые являются интерфейсом между записями и аппаратурой.

Конечным этапом создания нового проекта является компоновка объектных файлов и библиотек в единый исполняемый файл. dbd файлы помещаются в файловую систему, доступную из RTEMS и используются далее на этапе инициализации ИОС.

6.2 Процесс инициализации и старта ИОС

Процесс инициализации ИОС изображен на рисунке 6. После загрузки исполняемого файла управление получает BSP (board support package), который инициализирует контроллер, стартует операционную систему и передает управление процессу Init из библиотеки librtemsCom. Процесс Init инициализирует консоль, romfs - файловую систему, сетевую подсистему и запускает командный интерпретатор ИОС (shell). С этого момента дальнейшее управление ИОС может осуществляться оператором вручную через консоль (telnet) путем ввода команд. Более удобно использование командного файла, который запускается автоматически при старте shell. Имя файла и его расположение в файловой системе фиксировано – /romfs/st.cmd. Файл содержит команды интерпретатора, с помощью которых продолжается дальнейшая инициализация ИОС.

Загрузка базы данных из файловой системы в память контроллера является продолжением инициализации. Загрузка БД разделена на два этапа.

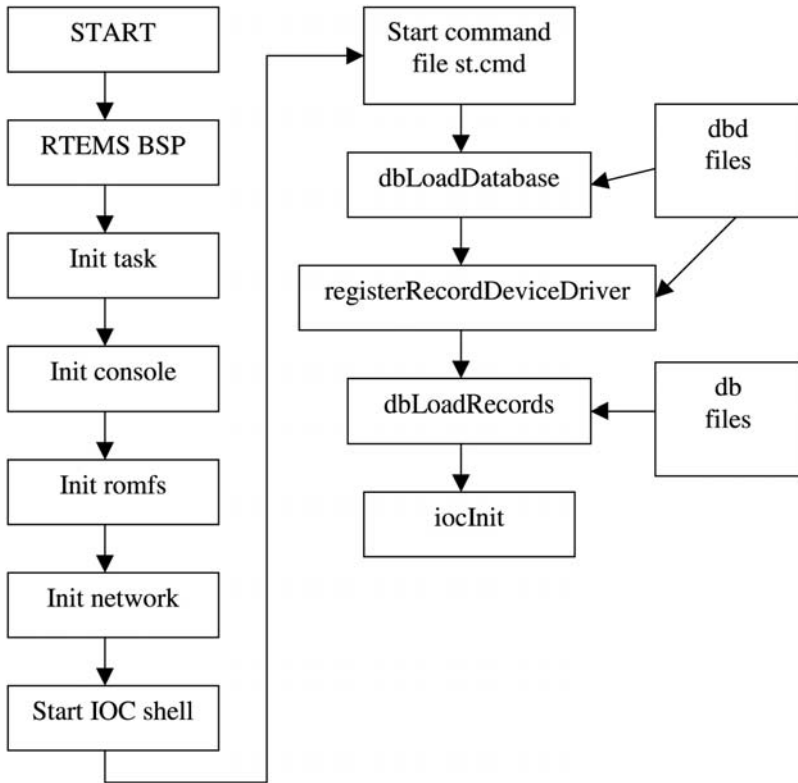


Рис. 6: Процесс инициализации IOC.

На первом этапе с помощью команды `shell dbLoadDatabase` интерпретируются `dbd` файлы с описаниями структур записей. Производится синтаксический анализ этих файлов. На этом этапе не создаются конкретных экземпляров записей, подготавливаются только структуры данных. Первый этап загрузки БД заканчивается вызовом команды `registerRecordDeviceDriver`. Данная команда регистрирует новые типы записей в IOC. Особенность IOC на базе RTEMS заключается в том, что файл `registerRecordDeviceDriver.c`, который включает в себе программную реализацию данной команды, создается динамически из тех же `dbd` файлов, которые содержат описания типов записей. Для этого служит Perl скрипт `registerRecordDeviceDriver.pl`.

Второй этап загрузки БД заключается в создании в памяти контроллера конкретных экземпляров уже известных типов записей. Это делается с помощью команды `dbLoadRecords`. Данная команда интерпретирует `db`-файлы, в которых разработчик БД описывает сколько и каких записей должно функционировать в системе. При описании конкретного экземпляра записи указываются также значения некоторых полей. Например, можно указывать ссылки на другие записи, частоту сканирования записей и т.д. Здесь же осуществляется привязка экземпляра записи к устройствам (поле `DTYP`).

После загрузки базы данных в память контроллера ИОС готов к старту. Старт осуществляется командой `iosInit`.

6.3 Дополнительные компоненты RTEMS

Для нормального функционирования ИОС понадобилась разработка дополнительных компонентов RTEMS - файловой системы `romfs` и сервера `telnet`.

Доступ к файловой системе необходим во время загрузки базы данных. При использовании `VxWorks` файлы БД хранятся на `host` системе, доступ к ним осуществляется через сетевую файловую систему `NFS`. Для RTEMS была разработана поддержка `Linux` - совместимой файловой системы `romfs`. Файлы `romfs` располагаются в системной памяти контроллера, поэтому доступ к ним осуществляется максимально быстро. При загрузке и интерпретации базы данных это дает большие преимущества по сравнению с другими типами файловых систем. Создается `romfs` на `host` системе с помощью утилиты `genromfs` из поддиректории, содержащей файлы с базой данных.

Для дистанционного управления ИОС через сеть `TCP/IP` для RTEMS был разработан `telnet` сервер. Сервер перенаправляет консоль на удаленный терминал.

7 Испытательный стенд

Для переноса ИОС на КАМАК контроллер использовался испытательный стенд, который состоял из `host` системы под управлением `Linux`, КАМАК крейта с установленным КАМАК контроллером на базе `Motorola MCF5307` и блока АЦП `C0609`.

Была разработана программная поддержка блока АЦП в ИОС для доступа к нему из записей типа `ai` (`analog input`). Работоспособность ИОС проверялась путем обращения через сеть `TCP/IP` к полям базы данных.

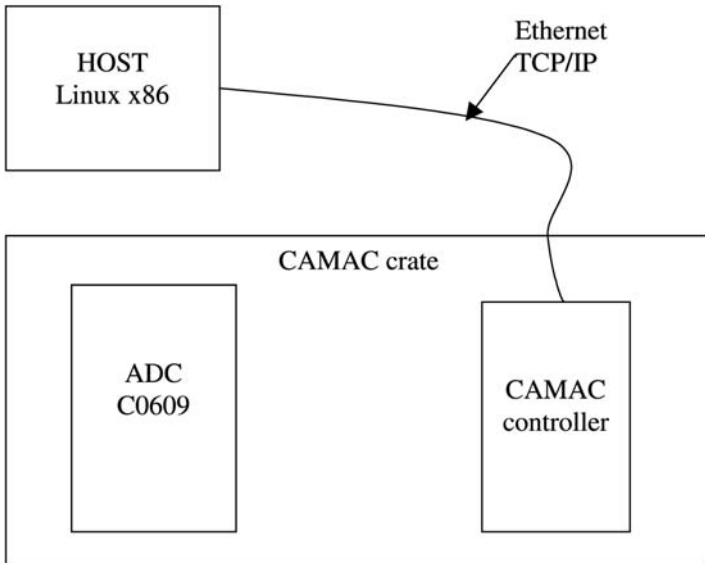


Рис. 7: Испытательный стенд.

8 Заключение

Проведенная работа по переносу компонентов EPICS на контроллеры CAMAC и VME показала возможность плавного перехода на современные системы сбора данных. Для построения EPICS системы можно использовать весь набор периферийных CAMAC блоков, разработанных в ИЯФ. Дешевизна разработанных контроллеров, а также использование бесплатного программного обеспечения, позволяет создавать EPICS системы с минимальными финансовыми издержками.

Список литературы

- [1] Martin R. Kraimer. EPICS Input / Output Controller (IOC) Application Developer's Guide.
- [2] Philip Stanley, Janet Anderson, Marty Kraimer. Record Reference Manual.
- [3] Jeffrey O. Hill. EPICS R3.12 Channel Access Reference Manual.
- [4] <http://www.aps.anl.gov/epics>.

В.Р. Мамкин

**Использование VME и SAMAC контроллеров
в составе EPICS**

V.R. Mamkin

**VME and SAMAC controllers employment
in the EPICS**

ИЯФ 2002-12

Ответственный за выпуск А.М. Кудрявцев
Работа поступила 20.02.2002 г.

Сдано в набор 22.02.2002 г.

Подписано в печать 25.02.2002 г.

Формат бумаги 60×90 1/16 Объем 1.4 печ.л., 1.1 уч.-изд.л.

Тираж 90 экз. Бесплатно. Заказ № 12

Обработано на IBM PC и отпечатано на
ротапринте ИЯФ им. Г.И. Будкера СО РАН

Новосибирск, 630090, пр. академика Лаврентьева, 11.