



Стандарты GPIB, 488.2 и SCPI и их влияние на развитие автоматизации измерений

А.Баженов (Лаборатория Atlas Electronic Group)

Данная публикация посвящена одному из наиболее успешных стандартов в области автоматизации измерений: General Purpose Interface Bus (GPIB). Появившись более 30 лет назад, этот стандарт и поныне не утратил своих позиций. В статье анализируются причины его долголетия и перспективы развития его возможных приемников.

Родившийся более 30 лет назад и сделавший успешную карьеру в области автоматизации измерений, стандарт General Purpose Interface Bus (GPIB) и не думает уходить на покой. Не видно причин для его исчезновения, и, напротив, в каждом году появляются новые приборы с этим интерфейсом, растет номенклатура и число фирм-производителей, расширяется программная поддержка существующих решений.

Технические идеи, положенные в основу GPIB, оказались удачно сбалансированными. При маркетинговой поддержке гиганта приборостроения Hewlett-Packard стандарт GPIB оказал мощное влияние на рынок автоматизации. Достаточно назвать стандарты VXI, SCPI, VISA, PXI, IV.

Исторический обзор современного состояния

В 1965 году фирма Hewlett-Packard (HP) разработала интерфейс Hewlett-Packard Interface Bus (HP-IB) для связи приборов этой фирмы с компьютерами, а в следующем году выпустила свой первый компьютер. Он был разработан как многоцелевой приборный контроллер для растущего семейства программируемой контрольно-измерительной аппаратуры HP. Эта комбинация измерительной техники и компьютера дала пользователю принципиально новое качество работы с измерительными системами.

В HP-IB были заложены здоровые технические принципы: достаточно высокая скорость передачи, приемлемое число приборов на шине, гибкость топологии системы, достаточно большие расстояния между приборами. Несомненно, сказался большой опыт фирмы в данной области. Нововведение было хорошо воспринято промышленностью, и по прошествии ряда лет встал вопрос о принятии HP-IB как стандарта (Табл. 1).

1965	1975	1987	1990	1992	1993
HP-IB	HP-IB меняет название на GPIB и становится стандартом IEEE 488	Принят стандарт 488.2; IEEE 488-1978 становится IEEE 488.1	Принята спецификация SCPI	Ревизия IEEE 488.2	Быстродействующая версия 488.1

Табл. 1. История стандартов GPIB

Обеспечение надежной и хорошо описанной связи измерительной техники и компьютера было первым шагом в построении открытых измерительных систем. В конце 60-начале 70 годов был популярен другой недорогой интерфейс связи оборудования и компьютеров BCD (Binary-Coded Decimal). Интерфейс HP-IB получил большее распространение ввиду его полной открытости и документированности. В 1975 году он был переименован в GPIB и официально одобрен Институтом инженеров электротехников и электронщиков (IEEE) как стандарт IEEE 488-1975 [IEEE-488.1], который позднее стал стандартом ANSI/IEEE 488.1. Этот стандарт зафиксировал электрические и механические соединения между компьютерами и приборами.

Помимо HP, аппаратуру GPIB начали производить для рабочих станций, а затем и для IBM-совместимых ПК фирмы National Instruments (NI) и другие.

Компании Hewlett-Packard, Tektronix, National Instruments лидеры в производстве приборов, соответствующих более новым стандартам, например, VXI и PXI, расширяют и номенклатуру GPIB, и различных средств сопряжения GPIB с более новыми интерфейсами.

Чтобы не быть голословными, посмотрим в каталог или на сайт такой фирмы, как National Instruments и сопоставим количество поддерживаемой аппаратуры в популярных в России инструментальных средах LabView и LabWindows (Табл. 2, 3).

Год	Интерфейс прибора		
	VXI	GPIB	последовательный
1993	75	159	6
1999	109	411	40

Табл. 2. Число драйверов LabView для измерительных приборов, выполненных в разных стандартах по каталогам National Instruments

Год	Интерфейс прибора	
	VXI	GPiB
1993	10	21
1999	6	21

Табл. 3. Число интерфейсов с другими шинами

Здесь бросается в глаза значительно опережающий рост поддерживаемых GPiB-приборов по отношению к приборам VXI. Правда, объяснение этому простое: приборы VXI сразу создавались с поддержкой LabView/LabWindows, а для GPiB в 1993 году их просто не успели написать, что делалось в последующие годы.

При этом произошел принципиальный сдвиг в базовых платформах автоматизации измерительных стендов. В 1993 году более половины интерфейсов GPiB приходилось на рабочие станции Sun, SGI, IBM RISC System/6000 и HP. В 1999 году существенную роль играет поддержка мезонинов PCMCiA, PMC, PC/104, EXM и сетей Ethernet, Parallel, Firewire, USB, RS-232/485.

Интересно обратиться к номенклатуре типов приборов, поддерживаемых в разных стандартах. Приборы с интерфейсом GPiB доминируют в таких классах, как анализаторы-тестеры, системы калибровки, осциллографы и источники питания. Модульные решения (VXI) преобладают для многоцелевых систем, и самыми популярными приборами здесь являются всевозможные типы переключателей-мультиплексоров. Мультиметры в равной мере представлены в обоих случаях. Отметим, что приведенный анализ не вполне корректен. Фирма National Instruments и ее продукты относятся, скорее, к автоматизации лабораторных рабочих мест, чем к масштабным стендам. Сложными системами занимаются такие фирмы, как, например, HP, WaveTek, BK, Kinetic Systems и соотношение между классами продуктов у них другое, а LabView там не используется, поскольку программирование стендов требует программных средств другого уровня специальных языков типа ATLAS (Automated test language systems) и языков общего назначения типа АДА.

Не ставя перед собой задачу вступать в дискуссии по вопросу о перспективности того или иного стандарта, заметим только, что упомянутая фирма National Instruments, выпуская продукт LabView for Linux первоначально включила в него поддержку только (!) GPiB. Цитируем буквально: LabVIEW for Linux includes hardware drivers for National Instruments GPiB hardware. National Instruments GPiB hardware is the only hardware for which National Instruments provides Linux driver software. There are no current plans to port our other driver software to Linux. (<http://www.ni.com/linux>). На настоящий день (январь-2000) добавлена также поддержка VXI.

Другой пример в ежегодном обзоре IEEE Technology Survey 1999 в разделе Automation приводится такой аргумент бессмертия GPiB: всегда будут существовать задачи, требующие 1-2 приборов. Развитие технологий предъявляет все большие требования к средствам тестирования: современная технология беспроводной коммуникации требует тестирования токов утечки ниже 1 нА. Такие требования выполняются только в специальных приборах, но не в картах в компьютере.

Забегая вперед, можно назвать основные причины успеха GPiB. Это техническая состоятельность и рыночная привлекательность (Табл. 4).

Техническая состоятельность		Рыночная привлекательность
Интерфейс	Архитектура	
Простота и экономичность	Гибкость топологии системы	Полная документированность и открытость
Поддержка неискажённой передачи информации на аппаратном уровне	Достаточно высокая скорость передачи	Программная поддержка на уровне системы
	Число приборов в системе достаточно велико	
	Разумные расстояния между приборами	

Табл.4 Основные причины успеха GPiB

Обратим внимание в технических параметрах нигде нет слов дешевый и сверхпроизводительный. Успех GPiB это успех гармонично сбалансированной платформы.

В Табл. 5 даны сравнительные характеристики GPiB, модульных и распределенных решений по Tektronix [1].

Характеристика	GPIB	Модульные решения	Решения для умных преобразователей
Стандартизация	<ul style="list-style-type: none"> • IEEE, ANSI, IEC, British Std • Стандарты, использующие 88.1: 488.2 и VXI 	<ul style="list-style-type: none"> • IEEE K USAF • Последовательность спецификаций VXI-2 VXI-11 	<ul style="list-style-type: none"> • IEEE • Последовательность спецификаций 1451.1 - 1451.4
Популярность	<ul style="list-style-type: none"> • Отражает потребность связи компьютера и прибора. • Очень большое число производителей • Полная спецификация интерфейса для пользователей 	<ul style="list-style-type: none"> • Отражает потребность в модульных системах, управляемых компьютером. • Полная спецификация системы для производителей. • Медленно растущее число производителей 	<ul style="list-style-type: none"> • Отражает потребность в распределённых системах, управляемых компьютером • Полная спецификация системы для производителей • Новый стандарт
Стоимость интерфейса	<ul style="list-style-type: none"> • Низкая стоимость аппаратной части • Высокие затраты на разработку встроенного ПО • Высокая стоимость прибора (источник питания, механика) 	<ul style="list-style-type: none"> • Большая стоимость аппаратной части • Оправдывается при разработке серии приборов или в системе за счёт многократного использования разработки механики интерфейса и встроенного ПО 	<ul style="list-style-type: none"> • Низкая стоимость аппаратной части • Высокие затраты на разработку встроенного ПО • Низкая стоимость прибора (источник питания, механика)
Возможности	<ul style="list-style-type: none"> • До 15 приборов на шине 488.1 • Длина кабеля - до 20 м • Побайтная передача данных • Скорость передачи - 1Мбайт/с • Один спикер, множество слушателей • Прерывание от прибора • Приборной шины нет 	<ul style="list-style-type: none"> • До 13 физических приборов в 1 крейте • Максимальный размер объединительной панели - 19 дюймов • Передача данных по 8, 16, 24, 32 бит • Скорость передачи до 40 Мбайт/с • Компьютерная шина мастер-слейв • Прерывание от прибора - набор • Набор приборных шин • Протокол передачи сообщений схож с 488.2 	<ul style="list-style-type: none"> • Фактически нет ограничений на число приборов и расстояния между ними • Передача данных пакетами промышленных или информационных сетей • Скорость передачи - определяется сетью • Прерывание от прибора
Прибор	<ul style="list-style-type: none"> • Полнофункциональный прибор, способен работать автономно • Предназначены для интерактивного управления 	<ul style="list-style-type: none"> • Необходим крейт, контроллер слота 0, компьютер и ПО • Предназначены для работы в системе • Есть возможность непосредственной связи приборов 	<ul style="list-style-type: none"> • Необходим интерфейс с локальной сетью • Предназначены для интерактивного управления
Система	<ul style="list-style-type: none"> • Простая минимальная система из прибора, контроллера, ПО и кабеля • Система стандартизована только логически, через протокол 	<ul style="list-style-type: none"> • Необходим крейт, контроллер слота 0, компьютер и ПО • Система стандартизована с точки зрения механической, электрической, тепловой совместимости 	<ul style="list-style-type: none"> • Система из прибора, компьютера, ПО и кабеля • Система стандартизована только логически, через модель прибора
Работа прибора	<ul style="list-style-type: none"> • Данные и состояния прибора отображаются на его экране • Ручная работа • Интерактивная работа через программную переднюю панель 	<ul style="list-style-type: none"> • Данные и состояния прибора находятся в компьютере • Требуется ПО для работы • Интерактивная работа через программную переднюю панель 	<ul style="list-style-type: none"> • Данные и состояния прибора находятся в компьютере • Требуется ПО для работы • Интерактивная работа через программную переднюю панель

Табл.5 Сравнительные характеристики GPIB, модульных и распределенных решений

Что такое GPIB

Рассмотрение GPIB начнем с архитектуры системы, затем коснемся протоколов обмена, кратко рассмотрим аппаратный интерфейс 488.1 и программный интерфейс 488.2. Наибольшее время будет посвящено стандарту программирования SCPI.

В качестве конкретного примера рассмотрим вопросы программирования по стандартам 488.2 и SCPI прибора VXi с набором команд 488 и SCPI. При этом рассмотрим поддержку этих протоколов со стороны прибора и со стороны средства разработки на компьютере-контроллере.

В заключение перейдем к проблемам и недостаткам GPIB и SCPI и их влиянию на более новые стили программирования приборов VISA и IVI, а также стандарт IEEE 1451 для умных УСО.

Стандарт 488.1 определяет электрические и механические характеристики и модель коммуникации.

Топология сети может быть линейной и звездообразной (Рис. 1). В сети не может быть более 15 приборов, расстояние между ними не может превышать 4 м при полной длине кабеля до 20 м. Возможно расширение сети с помощью расширителей.

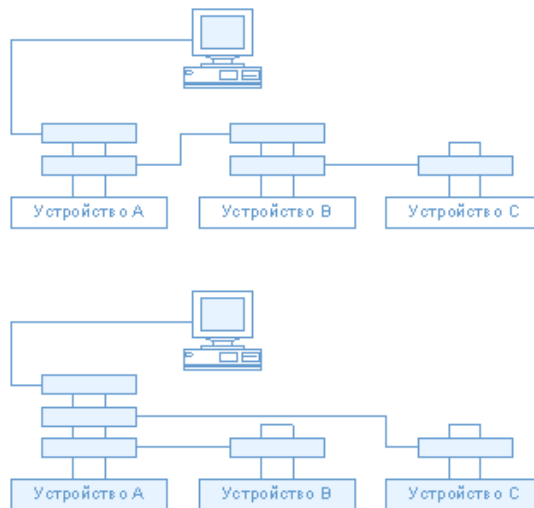


Рис.1 Архитектуры системы GPIB: линейная и звезда

Типы сообщений GPIB

Приборы GPIB обмениваются информацией, передавая сообщения интерфейса и сообщения, зависящие от прибора, через интерфейс системы.

Сообщения, зависящие от прибора, данные содержат информацию для прибора, такую, как инструкции программ, результаты измерений, статус, файлы данных.

Сообщения интерфейса команды управляют шиной. Выполняют функции инициализации, адресации, установки приборов в локальную или удаленную программную моду.

Типы приборов GPIB

Приборы GPIB могут быть Слушателями, Спикерами, Контроллерами. Спикер посылает данные одному или более Слушателям. Контроллер управляет шиной, посылая команды всем приборам.

Система GPIB обладает чертами и компьютера, и сети с централизованным управлением. Функции контроллера подобны функциям коммутатора телефонной сети. Контроллер следит за состоянием шины и при обнаружении запроса на передачу сообщения соединяет Спикера со Слушателем. Такое управление требуется не всегда, а в основном при смене Спикеров и Слушателей.

В системе GPIB может быть только один Системный контроллер. Он может передать функцию управления другому Контроллеру, сделав его Текущим контроллером.

Сигналы и линии GPIB

Интерфейс состоит из 16 сигнальных линий и 8 линий заземления или возврата сигнала. Сигнальные линии группируются следующим образом: 8 линий данных, 3 линии протокола и 5 линий управления интерфейсом. Назначение контактов разъема GPIB приведено в Табл. 6.

Мнемоника	Номер контакта	Номер контакта	Мнемоника
DIO1	1	13	DIO5
DIO2	2	14	DIO6
DIO3	3	15	DIO7
DIO4	4	16	DIO9
EOI	5	17	GND - EOI
DAV	6	18	GND - DAV
NRFD	7	19	GND - NRFD
NDAC	8	20	GND - NDAC
IFC	9	21	GND - IFC
SRQ	10	22	GND - SRQ
ATN	11	23	GND - ATN
SHIELD	12	24	GND - GND

Табл. 6. Назначение контактов разъема GPIB

Линии данных предназначены для передачи данных и команд. Все команды и большинство данных используют 7-битный набор ASCII, в котором 8 бит не используется или используется для четности.

Линии протокола управляют асинхронным образом передачей байтов между приборами. Используемый способ называется 3-проводным рукопожатием со взаимной блокировкой (interlocked handshake). Он гарантирует передачу данных по линиям данных без ошибок.

Линия NRFD (not ready for data) отображает, готов ли прибор к приему байта сообщения.

Линия устанавливается всеми приборами при приеме команд и Слушателями при приеме данных.

Линия NDAC (not data accepted) означает, приняты ли данные. Линия устанавливается всеми приборами при приеме команд, и Слушателями при приеме данных.

Линия DAV (data valid) сообщает, что линии сигналов и данных стабильны (правильны) и могут быть правильно восприняты приборами. Контроллер устанавливает линии DAV при посылке команд, а Спикер устанавливает эти линии при посылке данных.

Линии управления интерфейсом

ATN (attention) Контроллер устанавливает линию ATN в логическую 1 при посылке команд, и в логический 0, когда Спикер может посылать данные.

IFC (interface clear) Системный контроллер устанавливает линию IFC для инициализации шины с тем, чтобы стать Текущим контроллером.

REN (remote enable) Системный контроллер устанавливает линию REN при установке приборов в локальную или удаленную программную моду.

SRQ (service request) Любой прибор может установить линию SRQ для асинхронного запроса на обслуживание Контроллером.

EOI (end or identify) Имеет 2 назначения. Спикер использует линию для обозначения конца строки сообщения, а Контроллер таким образом сообщает приборам, что они должны идентифицировать свой ответ при параллельном опросе.

Стандарт 488.2

Одновременно с принятием стандарта 488.1 фирмы HP, Tektronix и другие начали работать в рамках IEEE над первым уровнем программной стандартизации, получившей имя IEEE-488.2 Standard Codes and Formats (Стандартные коды и форматы, принят в 1987 году) [IEEE-488.2]. Такая стандартизация была существенным шагом вперед, обеспечивая надежный способ общения по шине в смысле механизма доставки команд, однако относительно самих команд попытки стандартизации сделано не было. В качестве аналогии для этой ситуации можно привести пример телефонной системы, в которой получено хорошее качество связи, но с абонентами, говорящими на разных языках.

Разработка стандарта шла поэтапно. В 1987 году был выпущен набросок стандарта, а старое название 488 преобразовано в 488.1. В 1990 году был принят стандарт SCPI для приборов с интерфейсом 488. В 1992 году была принята новая ревизия стандарта 488.2.

Цель стандарта 488.2 преодоление различных ограничений исходного стандарта 488. Первичный стандарт

не определял форматы данных, общие команды конфигурации, протокол обмена сообщениями, и команды, специфичные для приборов. Стандарт 488.2 не затрагивает аппаратной части, основываясь на 488.1.

Основной концепцией 488.2 является точная речь и снисходительные слушатели. Стандарт определяет требования к Контроллеру и приборам с тем, чтобы получить высоконадежную и эффективную систему. Что такое снисходительные слушатели? Приборы, удовлетворяющие 488.2, должны уметь правильно интерпретировать команды и их небольшие вариации. В частности, они должны производить анализ содержимого регистров и правильно интерпретировать полную и краткую форму команд. Последнее обстоятельство существенно упрощает пользование системой. Например, установка частоты генератора может быть произведена любым из перечисленных ниже способов:

- Frequency 24.356
- Frequency +24
- FREQUENCY 24
- FREQuency 2.4E+1
- Frequency +2.43E+1

Существенно то, что интерпретация формата данных также возложена на прибор. Округление производится в соответствии с внутренней точностью прибора. При запросе FREQ? Прибор возвратит установку в одном из предопределённом стандартом форматов, независимо от того, каким образом эта установка в нем была произведена.

Контроллеры 488.2

Управляющие последовательности. В стандарте 488.2 определены управляющие последовательности (Табл. 7), которые формируют точные сообщения 488.1, выдаваемые Контроллером, а также порядок сообщений, если их требуется послать несколько. Всего определено 19 последовательностей 15 обязательных и 4 опциональных.

Описание	Управляющая последовательность	Требования
Посылка команд ATN-истинно	Send Command	Обязательно
Установка адреса для посылки данных	Send Setup	Обязательно
Посылка команд ATN-ложно	Send Data Bytes	Обязательно
Посылка программного сообщения	Send	Обязательно
Установка адреса для получения данных	Receive Setup	Обязательно
Получение данных ATN-ложно	Receive Response Message	Обязательно
Получение сообщения ответа	Receive	Обязательно
Активизация линии IFC	Send IFC	Обязательно
Очистка приборов	Device Clear	Обязательно
Установка приборов в автономное состояние	Enable Local Controls	Обязательно
Установка приборов в состояние удалённого управления	Enable Remote	Обязательно
Установка приборов в режим удалённого управления в состоянии локаута	Set RWLS	Обязательно
Установка приборов в автономное состояние в состоянии локаута	Send LLO	Обязательно
Чтение байта статуса 488.1	Read Status Byte	Обязательно
Посылка сообщения выполнения триггера группе (GET)	Trigger	Обязательно
Передача управления другому прибору	Pass Control	Обязательно
Параллельный опрос	Perform Parallel Poll	
Конфигурация приборов для параллельного опроса	Parallel Poll Configure	
Отмена возможности параллельного опроса	Parallel Poll Unconfigure	

Табл.7 Управляющие последовательности 488.2

Применение стандартных последовательностей устраняет неопределенности на шине и обеспечивает надежную связь. Причина возможных неопределенностей реализация функций управления шиной аппаратным способом без организации конечного автомата.

Протоколы 488.2.

Протоколы (Табл. 8) объединяют наборы управляющих последовательностей, с тем, чтобы выполнить полную измерительную операцию. Определено 2 обязательных и 6 опциональных протоколов. Протокол RESET обеспечивает инициализацию всех приборов. Протокол ALLSPOLL опрашивает каждый прибор последовательно и возвращает байт статуса каждого прибора. Протоколы PASSCTL и REQUESTCTL обеспечивают передачу управления шиной разным приборам. Протокол TESTSYS реализует функцию самотестирования каждого прибора.

Ключевое слово	Название	Требования
RESET	Установка системы	Обязательно
FINDRQS	Поиск прибора, требующего обслуживания	Опционально
ALLSPOLL	Последовательный опрос приборов	Обязательно
PASSCTL	Передача управления	Опционально
REQUESTCTL	Запрос управления	Опционально
FIDLSTN	Поиск слушателей	Опционально
SETADD	Установка адреса	Опционально, но требует FIDLSTN
TESTSYS	Автотест системы	Опционально

Табл.8 Протоколы контроллера 488.2

Протоколы FINDLSTN и FINDRQS поддерживают управление системой GPIB. При этом используются возможности, заложенные в стандарте 488.1. Контроллер выполняет протокол FINDLSTN, генерируя адрес Слушателя и проверяя наличие прибора на шине по состоянию линии NDAC. Протокол FINDLSTN возвращает список Слушателей, и выполнение этого протокола до начала работы прикладной программы гарантирует правильность текущей конфигурации системы. Для работы протокола FINDRQS используется возможность проверки линии SRQ. Входной список устройств можно ранжировать по приоритетам. Тем самым обеспечивается обслуживание наиболее ответственных приборов в первую очередь.

Приборы 488.2

По сравнению со стандартом 488.1, спецификация 488.2 требует серьезной поддержки встроенного программного обеспечения приборов, а в некоторых случаях расширения аппаратной части. Однако для конечного пользователя программирование становится существенно проще, поскольку приборы поддерживают стандартный набор команд и запросов предопределенным способом, используя стандартный протокол обмена сообщениями и форматы данных. Протокол обмена сообщениями 488.2 является базой для стандарта SCPI языка управления приборами.

Стандарт 488.2 определяет минимальный набор возможностей интерфейса 488.1 для приборов. Все приборы должны иметь возможность посылать и принимать данные, запрашивать обслуживание и отвечать на сообщение очистки устройства. Также определен формат команд, посылаемых в приборы, и формат

ответов на запросы, выдаваемых приборами. Поскольку эти команды и запросы одинаковы для всех устройств, они стандартизованы. Стандартизована также и статусная информация.

В табл. 9 приведены обязательные команды стандарта 488.2

Мнемоника	Группа	Описание
*IDN?	Данные системы	Запрос идентификации
*RST	Внутренняя операция	Сброс
*TST	Внутренняя операция	Запрос на самотестирование
*OPC	Синхронизация	Завершение операции
*OPC?	Синхронизация	Запрос завершения операции
*WAI	Синхронизация	Ожидание завершения
*CLS	Статус и событие	Очистить статус
*ESE	Статус и событие	Разрешить статус события
*ESE?	Статус и событие	Запрос разрешения статуса события
*ESR?	Статус и событие	Запрос разрешения регистра статуса
*SRE	Статус и событие	Разрешить запрос сервиса
*SRE?	Статус и событие	Запрос разрешения запроса сервиса
*STB?	Статус и событие	Чтение байта запроса статуса

Табл.9 Обязательные команды стандарта 488.2

Информация, извлекаемая из статусного регистра, имеет определённый формат и предоставляет необходимые сведения о наличии и статусе устройств в системе, возможных ошибках, запросах и т.п.

Язык программирования приборов SCPI

Прояснив вопросы доступа к приборным интерфейсам, можно перейти собственно к вопросам программирования приборов. Классическим способом здесь является программирование с помощью посылок строк ASCII. В 60-80-х годах разработчики ориентировались на способ программирования приборов, подобный тому, как это делалось в современных им средствах разработки в условиях дороговизны памяти. Например, программные функции вызывались как F1, F2, и т.д., диапазоны измерений как R1, R2, и т.д. При этом возникали трудности для охвата функциональности из-за недостатка алфавита. Даже в одной и той же фирме для разных изделий использовались разные команды для одинаковых функций. Подобный стиль получил на Западе название R2D2.

Неудачной попыткой стандартизации языка команд, предпринятой BBC США, был язык CIL (Command Instrument Interface Language). В 1990 году был предложен язык SCPI (Standard Commands for Programmable Instruments) [SCPI] как открытый стандарт, который определяет общий набор команд для программируемых устройств.

Этот стандарт состоит из нескольких частей. Основные из них следующие:

- Синтаксис;
- Стандартные команды;
- Форматы данных.

Синтаксические правила определяют стандарты на построение лексем. По существу, они сводятся к обеспечению механизма посылки сложных многоуровневых команд и получению запросов, а также передаче параметров. Эта часть, как и описание форматов передачи данных, носит чисто технический характер.

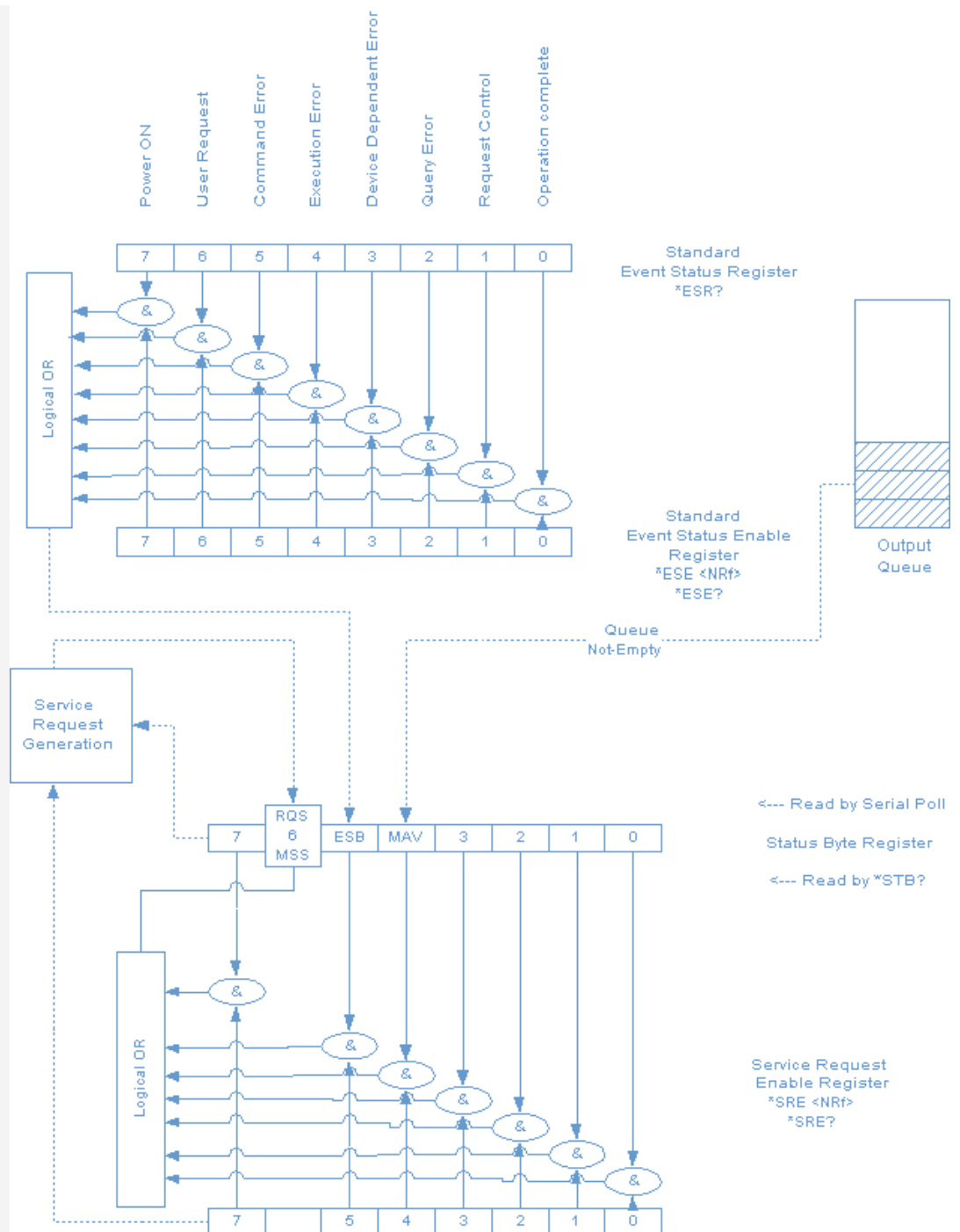


Рис.2 Модель отчёта о состоянии

Наиболее важной частью стандарта является модель универсального программируемого прибора. На этой основе и строятся команды управления, которые объединены в более чем 20 групп.

Модель прибора SCPI.

Как уже говорилось, SCPI не просто язык. Для достижения совместимости и классификации групп команд, SCPI определяет модель программируемого прибора. Общая модель [SCPI], представленная на рисунке 3, применима ко всем типам измерительных приборов.

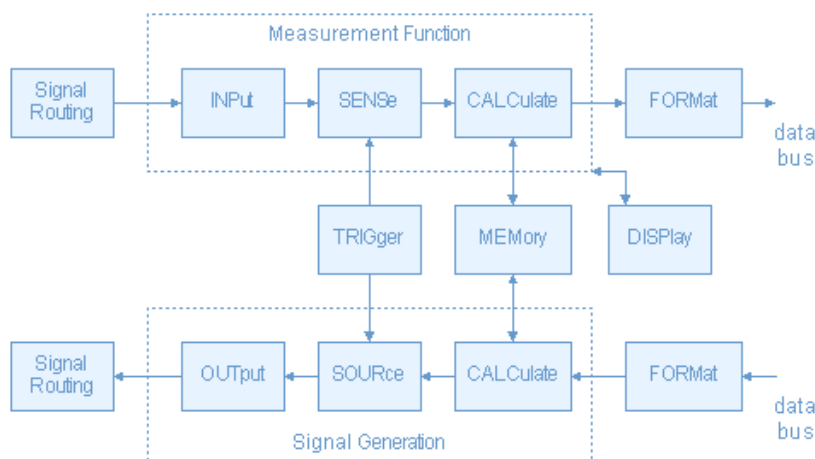


Рис.3 Модель программируемого прибора

В стандарте SCPI реализованы три важных идеи совместимости команд управления: горизонтальная, вертикальная и функциональная. Эти идеи удобно проиллюстрировать на примере. Предположим, что контроллер посылает команды различным осциллографам для измерения напряжения. Эти команды будут иметь один и тот же вид: SENSE1:VOLT:RANGE 1.0,

То есть в SCPI используются одни и те же команды для одинаковых функций внутри семейства приборов вертикальная совместимость. Реализация самих функций возложена на разработчика программного обеспечения прибора.

SCPI реализует команды различного уровня: от высокоуровневых команд, которые легко запомнить и использовать, до низкоуровневых, которые адресуют конкретику функциональности и обеспечивают тонкую настройку. Высокоуровневая команда MEASure:FERQuency? или ее короткая форма MEAS:FERQ? может быть послана различным приборам. Например, частотомеру или счетчику, которые выполняют такое измерение по-разному. Использование одних и тех же команд для приборов разного типа называется горизонтальной совместимостью.

Третья форма совместимости команд называется функциональной совместимостью. Она предполагает соответствие одинаковых команд одинаковым функциям разных приборов. Например, если и анализатор спектра, и генератор могут качать частоту, и если команды частоты и качания используются в обоих приборах, они будут функционально совместимы для данного применения.

Конкретный прибор может содержать не все представленные в обобщенной модели компоненты. В языке SCPI для каждого функционального компонента определяется иерархический набор команд управления конкретными особенностями.

Например, вольтметр, имеющий один вход и не содержащий буферной памяти, может включать функции измерения, блоки триггера и формата.

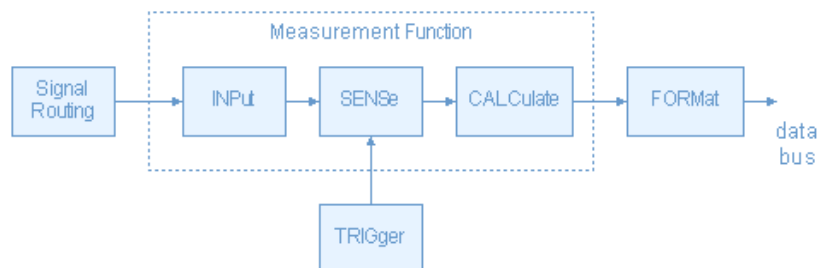


Рис.4 Упрощенная модель вольтметра

Подобным же образом генератор с одним выходом и без триггера, содержит блоки генерации и формата.

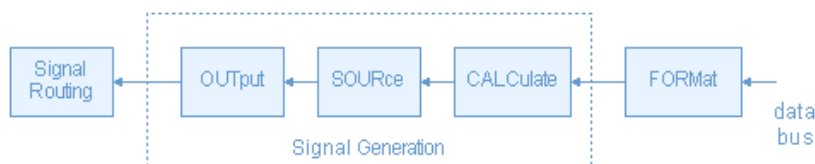


Рис.5 Упрощенная модель генератора

Наличие в приборе функции не обязательно означает, что должна быть соответствующая команда. Это бывает в тех случаях, когда данный блок имеет фиксированную конфигурацию, которая не конфликтует с соответствующими командами и имеет фиксированные значения по умолчанию после перезапуска *RST. Например, прибор может иметь фиксированный триггер по уровню ТТЛ, без настраиваемых параметров. В таком случае блок триггера и команды не нужны.

Компонент сопряжения с объектом (signal routing) связывает прибор с внешним сигналом; компонент измерения преобразует сигнал в промежуточную форму; компонент генерации сигнала преобразует данные в реальные сигналы. Компонент памяти хранит данные в приборе. Компонент формата преобразует данные прибора в формат данных шины. Компонент триггер синхронизирует операции прибора с внутренними

функциями, внешними событиями или другими приборами.

Функция измерения предоставляет наиболее высокий уровень совместимости приборов, поскольку измерение зависит от сигнала, а не от функциональности приборов. В большинстве случаев можно заменить один прибор другим для проведения конкретного измерения без изменения команд SCPI.

Функции измерения и генерации сигнала распадаются на 3 компонента. Блоки INPUT и OUTPUT служат для связи внешнего сигнала с измерением или генерацией. Блоки SENSE и SOURCE преобразуют сигнал во внутренний формат до его преобразования в данные и наоборот. CALCulate связывает данные прибора с данными приложения.

Перечислим для справки пункты раздела Модель прибора с небольшими комментариями (Табл. 10.).

№	Пункт	Комментарий
2.1	Signal routing	ROUTE блок
2.1.1	Setting the destination for Data Flow	Связана с FEED в блоке Internal routing
2.2	Measurement function	Разделение на INPUT, SENSE, CALCulate не определено в "Модели прибора" в силу того, что на этом низком уровне разделения реализация становится приборно-зависимой и нет горизонтальной совместимости. Например, измерение частоты сигнала делается счетчиком в блоке SENSE, а анализатором спектра в блоке CALCulate. В данном случае горизонтальная совместимость восстанавливается при использовании высокоуровневой команды MEASure. Функции согласования сигнала до преобразования в данные в блоке SENSE такие, как фильтры, смещение, перенос частоты, ослабление.
2.2.1	INPUT	Преобразование во внутренние "сырые" данные. Команды преобразования, например такими, как диапазон, разрешение, время экспозиции. Этот блок является обязательным.
2.2.2	SENSE	Преобразование "сырых" данных в более удобный для приложения формат, что обычно включает единицы измерения и обработку, например, вычисление времени нарастания по форме импульса.
2.2.3	CALCulate	Ситуация с генерацией сигналов (SOURCE) подобна и решается аналогично с блоком MEASure
2.3	Signal Generation	
2.3.1	OUTPUT	
2.3.2	SOURCE	
2.3.3	CALCulate	
2.4	TRIGGER	Синхронизация с внешними событиями
2.5	MEMORY	Управление данными внутри прибора. Причем данные могут быть как недоступными пользователю (например, внутренние калибровки прибора), так и предназначенными для использования.
2.6	FORMAT	Основан на форматах данных 488.2. Примером является преобразование внутреннего формата в ASCII для общения с внешним компьютером.
2.7	Internal Routing	Связь блоков внутри прибора
2.7.1-7	Data and Control Flow, Numeric Suffixes, Lamina and cloned models	Детали реализации связей внутри приборов. Например: 2.7.3. Lamina and cloned models Клонированная модель подходит для экстенсивного расширения базовой модели посредством простого клонирования несвязанных блоков. Например, многоканальный осциллограф с независимыми каналами

Табл. 10 Компоненты модели прибора SCPI

Другим удобством использования приборов, управляемых от компьютера, является возможность обмена данными. Поскольку SCPI стандартизует форматы обмена данных, не возникает проблемы их дополнительной интерпретации, например, при использовании горизонтально совместимых команд. В наиболее абстрактной форме это проявляется при представлении данных в виде файлов и обмене между приборами. Например, регистратор передает буфер данных анализатору спектра для частотного анализа, плоттеру для печати, компьютеру для отображения на экране дисплея.

Пример. В качестве иллюстрации рассмотрим уже знакомый по страницам МКА прибор для мониторинга ядерных объектов по сигналам от датчиков ядерного излучения и управления видеокамерами [МКА-97].

Прибор представляет собой 8-канальный таймер-счетчик с двумя выходами управления. Прибор выполнен в стандарте VXI и поддерживает протокол обмена сообщениями. В нем есть внутренняя память для установок и буфер данных. Поскольку прибор содержит микропроцессор (68000/68030), в нем есть соответствующая периферия: часы реального времени, последовательный порт, и т.п. Помимо этого, в прибор включены элементы системы бесперебойного функционирования: контроллер основного питания (берется с шины VXI), черный ящик для хранения информации о сбоях и восстановлении питания и т.п. Подробно прибор описан на сайте [VXI].

Функциональность прибора отражается в командах, пример использования которых иллюстрируется выдержками из листинга на рис. 6. Команды запроса не приводятся для экономии места, за исключением тех, которые имеют единственную форму.

```

* *****
* **** Modular soft, DT9150 base, PNPI 1996 ****
* *****
*
*          Дата модификации - 14.06.96 - Ю.Соболев
*
*          IEEE 488.2 special commands:
*
*          Блок установок - фрагмент
*+ '*TIM 1'      Установка времени
*+ '*DAT 1'      Установка даты
*+ '*DST n'      Установка даты и пуск часов реального времени*
*+ '*SEL i'      Выбор номера канала
*+ '*THR 1'      Установка порога счетчика для активации видео
*+ '*PDU 1'      Установка времени экспозиции текущего счетчика
*
*          Блок работы с буферной памятью - фрагмент
*+ '*RCR'        Чтение указателя данных в буфере
*+ '*RCP'        Чтение установок времени экспозиции для текущих
*+ '*RCS'        Чтение времени старта набора
*+ '*RCB'        Чтение времени снятия питания шины VXI
*+ '*RCU'        Чтение времени восстановления питания шины VXI
*
*+ '*CLM'        Очистка буферной памяти
*+ '*RDT 1'      Чтение данных по указателю, равному 1, для всех
*                счетчиков
*+ '*BFS'        Чтение размера свободной памяти в буфере
*
*          Блок работы с выводами - фрагмент
*+ '*VON'        Включить видеокамеру
*+ '*VOF'        Выключить видеокамеру
*
*          общие команды IEEE 488.2 - фрагмент:
*+ '*CLS'        Очистить все статусные регистры и флаги
*+ '*ESE'        Установить регистр ESE по маске пользователя
*+ '*IDN?'       Вывод названия фирмы, номеров модели и встроенного
ПО
*+ '*RST'        Установить прибор в исходное состояние
*
*          специальные команды SCPI:
*+ 'INST:NSEL i' аналог SEL i
*+ 'INST:NSEL?' аналог SEL?
*+ 'INST:STAT ON\OFF' включить\выключить текущий счетчик
*
*          общие команды SCPI - фрагмент:
*+ 'SYST:COMM:SER:REC:BAUD n'
*                установка скорости работы последовательного порта
*+ 'SYST:COMM:SER:REC:BITS n'
*                установка числа битов (7 или 8) последовательного порта
*+ 'SYST:COMM:SER:REC:SBIT n'
*                установка числа стоп-битов (1 или 2) последовательного
порта
*+ 'VXI:CONFigure:LADdress?'
*                запрос логического адреса VXI

```

Рис 6. Фрагмент листинга программы управления прибором на языке SCPI

Работа с прибором со стороны контролера шины VXI велась с помощью библиотеки NI-VXI и иллюстрируется фрагментом программы COLLECT (Ю.Соболев-1997): [Collect], приведенным на Рис.7.

```

INT16 WSwrt(INT16 la, UINT8 *buf, UINT32 count, UINT16 mode,
            UINT32 *retcount)
/*

фрагмент процедуры записи команды в прибор

*/

strcpy(vximgq,buf);

/* установка времени
if(strstr(vximgq,"*tim")!=NULL){
    GetLocalTime(&stime);
    i=strlen(vximgq);
    s[0]=vximgq[9]; s[1]=vximgq[10]; s[2]=0; stime.wSecond=atoi(s);

```

```

s[0]=vximsg[7]; s[1]=vximsg[8]; s[2]=0; stime.wMinute=atoi(s);
s[0]=vximsg[5]; s[1]=vximsg[6]; s[2]=0; stime.wHour=atoi(s);
SetLocalTime(&stime);
} else
/* установка даты
if(strstr(vximsg,"*dat")!=NULL){
GetLocalTime(&stime);
j=strlen(vximsg);
s[0]=vximsg[9]; s[1]=vximsg[10]; s[2]=0; stime.wYear=atoi(s);
if(stime.wYear<95)stime.wYear+=2000; else stime.wYear+=1900;
s[0]=vximsg[7]; s[1]=vximsg[8]; s[2]=0; stime.wMonth=atoi(s);
s[0]=vximsg[5]; s[1]=vximsg[6]; s[2]=0; stime.wDay=atoi(s);
SetLocalTime(&stime);
} else
/* выбор канала прибора
if(strcmp(vximsg,"*sel 1")==0)nse1=0; else
if(strcmp(vximsg,"*sel 2")==0)nse1=1; else
if(strcmp(vximsg,"*sel 3")==0)nse1=2; else
if(strcmp(vximsg,"*sel 4")==0)nse1=3; else
if(strcmp(vximsg,"*sel 5")==0)nse1=4; else
if(strcmp(vximsg,"*sel 6")==0)nse1=5; else
if(strcmp(vximsg,"*sel 7")==0)nse1=6; else
if(strcmp(vximsg,"*sel 8")==0)nse1=7;
return 0;
}

INT16 WSrd(INT16 la, UINT8 *buf, UINT32 count, UINT16 mode,
UINT32 *retcount)

/*

фрагмент процедуры чтения из прибора

*/

if(strcmp(vximsg,"*nrb?")==0){
i=sdatt.nrb; sprintf(buf,"%05d",i);
} else
if(strcmp(vximsg,"*nfb?")==0){
i=(int)sdatt.nfb; sprintf(buf,"%05d",i);
} else
if(strcmp(vximsg,"*pdu?")==0)sprintf(buf,"%05d",pdu);
else
if(strcmp(vximsg,"*fti?")==0)sprintf(buf,"%05d",fti);
else
if(strcmp(vximsg,"*ftl?")==0)sprintf(buf,"%05d",ftl);
else
if(strcmp(vximsg,"*thr?")==0)sprintf(buf,"%012d",thr[nse1]);
else
if(strcmp(vximsg,"*tim?")==0){
GetLocalTime(&stime);
sprintf(buf,"%02d:%02d:%02d:%02d",stime.wHour,stime.wMinute,
stime.wSecond,stime.wMilliseconds/10);
} else
if(strcmp(vximsg,"*dat?")==0){
GetLocalTime(&stime);
sprintf(buf,"%02d-%02d-%02d",stime.wDay,stime.wMonth,
stime.wYear%100);
} else
if(strstr(vximsg,"*lst")!=NULL){
strcpy(buf,"");
for(i=0;i<8;i++)if(cs&cmask[i])strcat(buf,"0"); else
strcat(buf,"X");
i=wSlot[la];
sprintf(str,"%05d",wCurrGateTime[i]); strcat(buf,str);

```

```

sprintf(str,"%05d",5000-sdat.nrb); strcat(buf,str);
strcat(buf,"1");
if(trgon)strcat(buf,"1"); else strcat(buf,"0");
if(upson)strcat(buf,"1"); else strcat(buf,"0");
strcat(buf,"1");
} else
strcpy(vximg,"");
return 0;
}

```

Рис 7. Пример использования библиотеки NI-VXI для работы с прибором со стороны шины VXI

Проблемы GPIB и SCPI. Развитие идей в VISA, IVI, IEEE 1451

Стандарт SCPI это полноценная реализация языка программирования приборов, которая с начала 90-х годов получила широкое распространение и в настоящее время доминирует на рынке как наиболее распространенный способ поддержки измерительных приборов. Он активно применяется для аналоговых приборов, особенно для сектора радиочастотных и микроволновых устройств. Стандарт SCPI позволяет работать на языке высокого уровня, а также с популярными графическими языками NI LabView и HP VEE.

Как видно из приведенного примера, работа пользователя с прибором, поддерживающим стандарты 488 и SCPI, очень проста не нужны обращения по физическим адресам внутри прибора и разбора тонкостей работы протокола обмена. Достаточно сформировать команду в виде строки ASCII и передать ее с помощью интерактивного средства или, в программной реализации, через процедуру в прибор. Такая простота позволяет работать с приборами персоналу, имеющему базовые навыки пользования компьютером и программирования на языке Basic.

Здесь можно перейти к проблемам и недостаткам GPIB и SCPI. Перефразируя известное высказывание Нильса Бора о простых и глубоких истинах, можно сказать, что неудачные решения плохи сами по себе, а в удачных недостатки являются следствием их достоинств. Стандарт GPIB это второй случай.

Наиболее очевидные недостатки GPIB:

- низкая скорость передачи данных;
- относительно высокая стоимость интерфейса;
- необходимость поддержки встроенного программного обеспечения 488.x

Наиболее очевидные недостатки SCPI:

- очень громоздкий стандарт;
- необходимость поддержки встроенного программного обеспечения, требующего лексического интерпретатора;
- медленная скорость передачи.

Рассмотрим упомянутые проблемы GPIB подробнее.

Базовая скорость передачи составляет 1 Мбайт/с (488HS 8 Мбайт/с). Для большинства полнофункциональных приборов, которые самостоятельно производят обработку информации, большая скорость и не нужна. Например, газоанализатор, в котором анализ масс остаточных газов построен на времяпролетной спектроскопии, имеет поток входной информации 50-200 Мбайт/с при оцифровке с частотой 5-10 нс и 8-12-разрядной точности измерения амплитуды сигнала [масс-спектрометр]. Обработанные же данные представляют спектрограмму величиной несколько Кбайт, а результат элементного анализа несколько десятков байт. Конечно, если необходим быстрый обмен между приборами, лучше подходят крейтовые реализации.

Стоимость интерфейса 488 в середине 70-х годов была очень велика. Например, Tektronix приводит такие цифры: в 1975 году цифровой мультиметр стоил \$3000 без интерфейса и \$3975 с интерфейсом. В настоящее время цена аппаратного интерфейса 488 составляет около \$30, что приемлемо для сложных приборов, но велико, чтобы ставить в тестер.

Необходимость поддержки встроенного программного обеспечения 488.2 наиболее важный аспект для производителя приборов. Помимо аппаратного интерфейса, в прибор необходимо поставить процессор, оперативную память для программ и написать лексический интерпретатор. Перед этой задачей спасовала вся приборостроительная промышленность СССР. Необходимы были недорогие процессоры для установки в приборы и средства разработки встроенного программного обеспечения. Ни того, ни другого отечественная промышленность не выпускала, как впрочем, и приемлемых по цене рабочих станций и ПК. Поэтому приборы шли с интерфейсом КОП (488.1) без поддержки 488.2. Заметим, что требования к памяти достаточно существенны: скомпилированный весьма ограниченный по набору команд (менее 100) интерпретатор в примере с процессором 680х0 занимает около 20 Кбайт.

Что касается SCPI, то здесь проблемы производителя возрастают еще больше. Разработка и поддержка полноценного интерпретатора SCPI выглядит осмысленной при выпуске серии приборов. Не случайно, что даже крупные производители измерительной аппаратуры часто ограничиваются набором команд 488.2 даже для приборов стоимостью более \$5000.

В одном ряду со сложностью интерпретатора и требуемого для его работы объема ОЗУ (RAM) стоит скорость

передачи команд SCPI. Передача лексем очень накладна и зачастую не годится для передачи данных. Например, в случае VXI, при сырой скорости передачи в несколько Мбайт/с, скорость протокола обмена сообщениями составляет 20-200 Кбайт/с. В результате появились такие решения, как скомпилированные команды, которые, по существу, вводят дополнительный слой кодировки-декодировки команд, но такие решения являются уже частнофирменными. В случае VXI есть выход в виде использования разделяемой памяти (VXI shared memory), в случае GPIB такой возможности нет.

В начале статьи уже приводились сведения о поддержке приборов с интерфейсом GPIB в популярных пакетах лабораторной автоматизации LabView/LabWindows. Приведенный в тексте пример использования библиотеки NI-VXI соответствует стилю программирования приборов середины 90-х годов. На смену этой технологии пришла новая технология с так называемыми приборными драйверами, о которой кратко говорилось в предыдущей, обзорной статье МКА, 4 за 1999 год [МКА-99]. Эта технология опирается на библиотеку VISA, которая берет на себя все проблемы доступа к аппаратной части.

По существу, драйверы приборов представляют собой набор вызовов VISA, специфичных для конкретного прибора. Такой подход впервые появился для приборов VXI в виде спецификации VXIplug-and-play. В настоящее время произведено обобщение этой идеологии на аппаратно-независимые классы приборов IVI (Interchangeable Virtual Instruments).

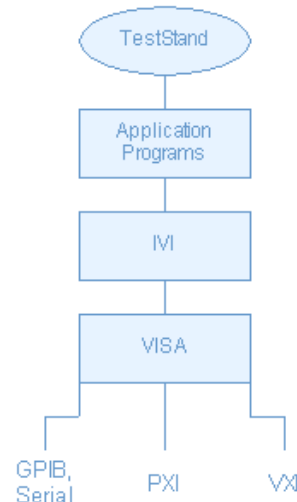


Рис.8 Этажерка программирования приборов

Строки SCPI могут поддерживать функциональность в общении с драйверами приборов. При наличии драйвера язык SCPI становится избыточным. Драйверы могут добираться до функций прибора различными способами. Если драйвер посылает команду SCPI (строку ASCII), она кодируется для передачи через интерфейс, декодируется прибором и вызывает соответствующую функцию. Если возможен прямой вызов функции, как в регистровых устройствах, соответствующую операцию можно выполнить гораздо быстрее, что существенно для быстрых операций. Тем не менее, большое количество приборов, по-прежнему, доступно только через набор команд SCPI.

Другим наследником идей модели прибора SCPI является идеология построения систем умных устройств сопряжения с объектом (УСО). Как и GPIB, она родилась в недрах Hewlett-Packard [1451]. На сей раз речь идет о промышленной автоматизации на основе инфраструктуры локальных сетей. Ввиду того, что промышленная автоматизация имеет многоуровневую структуру и для нее проблема сопряжения с объектом имеет чисто вспомогательный характер, в отличие от измерительных стендов, есть смысл выдавать на уровень автоматизации технологического процесса уже законченное измерение, как и в случае SCPI. В силу того, что за последние 10 лет принципиально изменились как стоимостные характеристики встроенного интеллекта и его поддержки, так и инфраструктура промышленного предприятия из-за повсеместного распространения локальных сетей, есть возможность внедрить интеллект буквально в каждый датчик/управляющий элемент или их группу, и придать каждому некий уникальный идентификатор и необходимые для проведения законченного измерения калибровки, параметры скорости опроса, и т.п.

Для описания УСО вводится универсальная модель, специфика отражается в наборе таблиц. Все эти данные располагаются на уровне локального интеллекта УСО, названного интеллектуальным модулем интерфейса с преобразователем STIM (Smart Transducer Interface Module). Доступ к локальной сети осуществляет другое устройство процессор приложения с сетевым интерфейсом NCAP (Network Capable Application Processor). Связь между STIM и NCAP стандартизована через универсальный интерфейс преобразователя TII (Transducer Independent Interface). Любопытно взглянуть как на модель STIM, так и на физический интерфейс TII (Рис. 9, Табл.11).

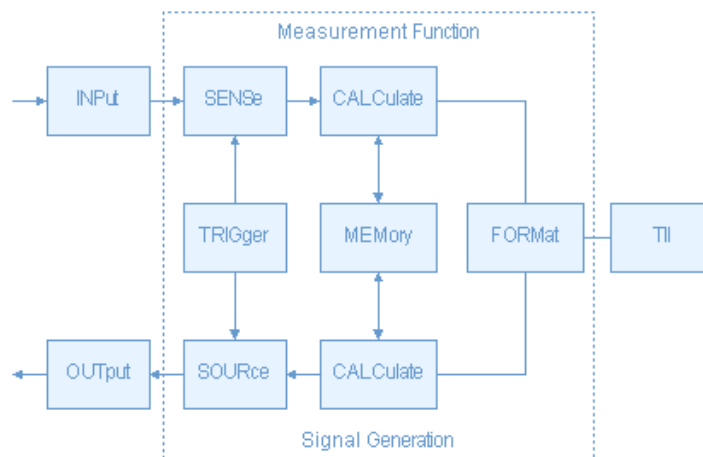


Рис.9 Интеллектуальный модуль интерфейса с преобразователем STIM (Smart Transducer Interface Module) в спецификации IEEE 1451

Группа	Линия	Описание
DATA	DATA_OUT	Передача данных от STIM к NCAP
	DATA_IN	Передача адреса и данных от NCAP к STIM
	DATA_CLOCK	Положительный срез - защелка данных для DIN и OUT
	N_IO_ENABLE	Означает, что идет передача данных и разделяет группы данных
Triggering	N_TRIGGER	Триггер (запуск)
Support	Power	5В
	COMMON	Земля
	N_ACKNOWLEDGE	1. Подтверждение триггера (запуска) 2. Подтверждение передачи данных
	N_STIM_DETECT	Используется NCAP для определения наличия STIM
Interrupt	N_IO_INTERRUPT NINT	Используется STIM для запроса обслуживания от NCAP

Табл.11 Интерфейс TII

В интерфейсе TII просматривается очевидное сходство с интерфейсом 488.1: физическая поддержка доставки данных, триггер, обеспечение питанием и прием прерывания. Отличие есть в количестве линий передачи данных, которых только две, что позволяет использовать как стандартные последовательные интерфейсы, так и решения под задачу.

В следующей публикации мы рассмотрим технологию VISA.

С автором можно связаться по e-mail: bazhenov@hep486.pnpi.spb.ru

Другие статьи раздела:

- 27.07.2009 [Методы кондуктивного охлаждения для защищённых компьютеров](#) (Серж Тиссо (Serge Tissot), Kontron Modular Computers SAS, МКА: мир ВКТ 6/2008)
- 05.02.2009 [Контроллеры ICP DAS на российском рынке автоматизированных систем](#) (Алексеев Д.Ю., ipc2U)
- 06.02.2008 [Стандарт nanoETXexpress: технология Embedded Technology eXtended продолжает развиваться](#) (Леонид Акиншин, ChipNEWS 9/07)
- 29.06.2006 [Лабораторные информационные системы на отечественном рынке](#) (А.Г. Терещенко, А.М. Янин (НИИ высоких напряжений))
- 06.06.2006 [Измерительная система температурного контроля генератора разработки ОАО "НИПС"](#) (ГАРКУША В.В., ГОЛОЛОБОВ В.И., ОАО "Новосибирский институт программных систем")

