

**РОССИЙСКИЙ ФЕДЕРАЛЬНЫЙ ЯДЕРНЫЙ ЦЕНТР
ВСЕРОССИЙСКИЙ НАУЧНО – ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ
ЭКСПЕРИМЕНТАЛЬНОЙ ФИЗИКИ
(РФЯЦ-ВНИИЭФ)**

На правах рукописи

УДК 539.1.07; 53.08:004; 001.89:004

Курякин Алексей Валерьевич

**АВТОМАТИЗАЦИЯ ФИЗИЧЕСКИХ ЭКСПЕРИМЕНТОВ НА ТРИТИЕВЫХ
КОМПЛЕКСАХ ИССЛЕДОВАТЕЛЬСКИХ УСТАНОВОК
«ТРИТОН», «АКУЛИНА» И «ПРОМЕТЕЙ»**

Специальность: 01.04.01 – приборы и методы экспериментальной физики

Диссертация на соискание ученой степени
кандидата физико-математических наук



Саров 2010

ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ И ОПРЕДЕЛЕНИЙ

ВНИИЭФ	– Всероссийский НИИ экспериментальной физики, г. Саров
ОИЯИ	– Объединенный институт ядерных исследований, г. Дубна
ЛЯР	– Лаборатория ядерных реакций им. Флерова (ОИЯИ)
ЛЯП	– Лаборатория ядерных проблем им. Дзелепова (ОИЯИ)
СПбГУ	– Санкт-Петербургский государственный университет
СПбГУТ	– Санкт-Петербургский государственный университет телекоммуникаций им. М.А. Бонч-Бруевича
НИИФ СПбГУ	– НИИ физики в СПбГУ
ТРИТОН	– исследовательская установка для изучения мюонного катализа ядерных реакций синтеза на фазотроне ЛЯП ОИЯИ
АКУЛИНА	– исследовательская установка для изучения экзотических, нейтронно-избыточных ядер на циклотроне U-400М ЛЯР ОИЯИ
ПРОМЕТЕЙ	– исследовательский стенд для изучения явлений пропускания и накопления изотопов водорода в металлах, ВНИИЭФ
АСКУ	– автоматизированная система контроля и управления
CRW-DAQ	– CuRves in Windows – for Data AcQuisition – созданный автором пакет для автоматизации АСКУ, http://www.crw-daq.ru
ПК	– персональный компьютер
ПО	– программное обеспечение
ОС, OS	– операционная система, operation system
ИК, IC	– ионизационная камера, ionization camera
АЦП, ADC	– аналого-цифровой преобразователь, analog - digital converter
МК, MCF	– мюонный катализ (ядерных реакций), muon catalyzed fusion
КМ	– конструкционные материалы
ИВ	– изотопы водорода
ЯВУ	– язык высокого уровня (C, Pascal, Fortran,...)
VM, VM	– Virtual Machine, Виртуальная Машина
RTDB	– Real Time Data Base – база данных реального времени
SCADA	– Supervisory Control And Data Acquisition
ШИМ	– широтно-импульсная модуляция
МНК	– метод наименьших квадратов
ROI	– region of interest, область интереса
JIT	– Just In Time, принцип компиляции «на лету»
DIM	– Distributed Information Manager, http://dim.web.cern.ch
МКИ	– метод концентрационных импульсов
MDI	– multi-document interface, многодокументный интерфейс
HTTP	– Hyper Text Transfer Protocol – сетевой протокол Internet

Оглавление

ВВЕДЕНИЕ	4
1. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ ИЗМЕРЕНИЙ НА УСТАНОВКЕ ВЫСОКОГО ДАВЛЕНИЯ ТРИТОН.....	12
1.1. Установка ТРИТОН для исследования мюонного катализа ядерных реакций.....	12
1.2. Структура тритиевого комплекса установки ТРИТОН	13
1.3. Аппаратурное и программное обеспечение АСКУ тритиевого комплекса.....	15
1.4. Инструментальный программный пакет для автоматизации физических экспериментов.....	17
1.4.1. Общая структура пакета	20
1.4.2. Компоненты пакета.....	22
1.5. Автоматизация комплекса подготовки газовой смеси и системы радиационного контроля	54
1.6. Управление сменными мишенями и измерение параметров газовой смеси.....	57
1.7. Анализ молекулярного состава газовой смеси.....	60
1.8. Результаты экспериментов на установке ТРИТОН	62
1.9. Выводы.....	65
2. УПРАВЛЕНИЕ СИСТЕМАМИ ДЛЯ ПОДАЧИ ИЗОТОПОВ ВОДОРОДА ЦИКЛОТРОНА У-400М И КОМПЛЕКСОМ ТРИТИЕВОЙ МИШЕНИ НА УСТАНОВКЕ АКУЛИНА.....	67
2.1. Установки АКУЛИНА.....	67
2.2. Управление системой подачи ИВ в ионный источник циклотрона У-400М	68
2.3. Управление комплексом тритиевой мишени.....	73
2.4. Результаты экспериментов на установке АКУЛИНА.....	80
2.5. Выводы.....	82
3. АВТОМАТИЗАЦИЯ ЭКСПЕРИМЕНТОВ НА ИССЛЕДОВАТЕЛЬСКОМ СТЕНДЕ ПРОМЕТЕЙ	83
3.1. Исследовательский стенд ПРОМЕТЕЙ	83
3.2. Аппаратура для автоматизации стенда ПРОМЕТЕЙ	91
3.3. Программное обеспечение для автоматизации экспериментов на стенде ПРОМЕТЕЙ	96
3.3.1. Управление газовакуумной системой	96
3.3.2. Управление нагревателями фильтров, источников и ловушек ИВ.....	99
3.3.3. Измерения на исследовательских ячейках.....	101
3.3.4. Масс-спектрометрические измерения	103
3.3.5. Система радиометрического контроля.....	105
3.4. Результаты экспериментов на стенде ПРОМЕТЕЙ.....	108
3.5. Выводы.....	109
ЗАКЛЮЧЕНИЕ.....	110
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	113
ПРИЛОЖЕНИЕ. ОСОБЕННОСТИ РЕАЛИЗАЦИИ ИНСТРУМЕНТАЛЬНОГО ПАКЕТА CRW-DAQ	120
1. Системы реального времени и отказоустойчивые системы	120
2. Примеры калибровок	127
3. Интерпретатор языка DaqScript.....	129
4. Реализация динамических расширений DPE.....	131
5. Построение отказоустойчивых систем управления.....	133
6. Работа АСКУ как навигация в фазовом пространстве.....	137
7. Распределенные АСКУ на базе DIM и WEB технологий	142
8. Конфигурирование измерительных систем CRW-DAQ.....	149
9. Система ограничения прав доступа к АСКУ	151
10. Особенности программной реализации пакета CRW-DAQ	153
11. Листинги	165

Введение

Работа выполнена в ходе создания исследовательских установок «ТРИТОН» и «АКУЛИНА» на ускорителях в Объединенном Институте Ядерных Исследований (ОИЯИ, г. Дубна) и исследовательского комплекса «ПРОМЕТЕЙ» в Российском Федеральном Ядерном Центре - Всероссийском НИИ Экспериментальной Физики (РФЯЦ-ВНИИЭФ, г. Саров).

Актуальность темы

В РФЯЦ-ВНИИЭФ интенсивно развивается тематика [1], связанная с изучением взаимодействия изотопов водорода (ИВ) с конструкционными материалами (КМ). Это обусловлено, в частности, потребностями развития современных направлений в энергетике – созданием ядерных реакторов нового поколения, созданием материалов и топливного цикла термоядерных реакторов, а также созданием материалов и инфраструктуры водородной энергетики [2].

В начале 90-х годов началось интенсивное научно-техническое сотрудничество между РФЯЦ-ВНИИЭФ и ОИЯИ в области исследований мюонного катализа (МК) [3] ядерных реакций синтеза и исследований структуры экзотических легких ядер и ядерных систем, находящихся на границе нейтронной стабильности. В этих исследованиях для получения пучков заряженных частиц и в качестве мишеней используются изотопы водорода и их смеси.

Экспериментальные и теоретические исследования МК ведутся в Лаборатории ядерных проблем (ЛЯП) ОИЯИ с 60-х годов. Здесь было открыто и объяснено явление резонансного образования молекулы $dd\mu$ [4], скорость которого сильно зависит от температуры, что обуславливает необычное свойство МК – зависимость ядерной реакции синтеза ИВ от температуры, плотности и состава смеси. В 1992 г. было предсказано [5], что скорость резонансного образования $dt\mu$ -молекулы в тройной H/D/T смеси намного выше, чем в бинарной D/T смеси.

Для исследований МК и нахождения оптимальных условий протекания МК в H/D/T-смесях в 1995 г. в низкофоновой лаборатории фазотрона ЛЯП ОИЯИ началось создание установки ТРИТОН [6], предназначенной для исследования процессов МК реакций синтеза в смесях ИВ в широком диапазоне температур, давлений и концентраций. По условиям эксперимента в мишени должно находиться

до 10 кКи ($3,7 \cdot 10^{14}$ Бк) трития, а в закрытом источнике - до 100 кКи [7]. Поэтому важной частью установки является автоматизированный радиационно-безопасный комплекс высокого давления для работ с тритием [8,9].

В 2000 г. в Лаборатории ядерных реакций (ЛЯР) ОИЯИ началась подготовка к экспериментам по получению и изучению нуклонно-нестабильных ядер ^4H и ^5H на установке АКУЛИНА [10,11]. Для этих экспериментов требовалось обеспечить пучок ускоренных ионов трития [12,13] и создать жидкую тритиевую мишень со стенками толщиной несколько микрон, в которой находится до 1 кКи ($3,7 \cdot 10^{13}$ Бк) трития [14,15], и систему контроля и управления для нее [16].

Задача создания радиационно-безопасных тритиевых комплексов и мишеней на установках ТРИТОН и АКУЛИНА решалась в РФЯЦ-ВНИИЭФ, где накоплен большой опыт работы с изотопами водорода.

Для изучения взаимодействия изотопов водорода с конструкционными материалами в 2001 г. в РФЯЦ-ВНИИЭФ создан исследовательский комплекс низкого давления ПРОМЕТЕЙ [17,18], позволяющий безопасно работать со всеми изотопами водорода, включая тритий. На нем с 2001 г. ведутся исследования проницаемости и сверхпроницаемости ИВ через металлические мембраны, а также явлений проникновения, накопления и диффузии водорода в металлах и конструкционных материалах.

Функционирование указанных исследовательских тритиевых комплексов и проведение на них физических экспериментов без высокого уровня автоматизации невозможно. Системы автоматизации тритиевых комплексов должны обеспечивать высокую надежность управления и измерения физических и технологических параметров. Кроме того, для безопасного проведения работ на этих комплексах требуется непрерывный мониторинг объемной активности трития в газовых коммуникациях и воздушной среде рабочих помещений, оповещение персонала при возникновении опасных ситуаций, а также наличие автоматических блокировок для предотвращения аварийных ситуаций.

Все эти обстоятельства обуславливают актуальность данной работы.

Цели и задачи работы

Целью диссертационной работы являлась автоматизация физических экспериментов на установках ТРИТОН, АКУЛИНА и ПРОМЕТЕЙ,

предназначенных для проведения фундаментальных и прикладных исследований с использованием изотопов водорода (в том числе трития) в ОИЯИ и РФЯЦ-ВНИИЭФ.

Для достижения этой цели были решены следующие задачи:

1. Создана инструментальная программная среда для быстрой разработки высоконадежных распределенных автоматизированных систем контроля и управления (АСКУ) физическими экспериментами на исследовательских установках, оснащенных газовакуумными комплексами для работы с изотопами водорода, включая тритий.
2. Разработаны алгоритмы и программное обеспечение для систем сбора данных и управления тритиевых комплексов и мишеней установок, предназначенных для проведения экспериментов по мюонному катализу ядерных реакций (ТРИТОН) и экспериментов по изучению легких нейтронно-избыточных ядер (АКУЛИНА) на ускорителях ОИЯИ.
3. Решена задача автоматизации исследований на стенде ПРОМЕТЕЙ, предназначенном для изучения взаимодействия изотопов водорода с конструкционными материалами в РФЯЦ-ВНИИЭФ, а также проводимых на нем физических измерений.

Научная новизна работы

Разработана оригинальная инструментальная программная среда (пакет CRW-DAQ), дающая качественно новые возможности для быстрой разработки высоконадежных многопоточных и многомашинных распределенных систем автоматизации физических измерений и управления для научных исследований в области тритиевых технологий. Пакет CRW_DAQ зарегистрирован в государственном реестре РФ [19].

Впервые созданы системы автоматизированного управления и сбора данных для тритиевых комплексов и мишеней, обеспечивающие в условиях неспециализированных лабораторий высокий уровень радиационной безопасности и надежности при проведении экспериментов с большими количествами трития (исследовательские установки ТРИТОН, АКУЛИНА и ПРОМЕТЕЙ).

Автоматизированный тритиевый комплекс установки ТРИТОН позволил впервые в экспериментах с D/T-смесями получить параметры цикла мюонного катализа dt-реакции в широком диапазоне температур 20-800 К, плотностей 0,2-1,2 LHD (плотность жидкого водорода) и концентраций трития 15-86%.

С использованием автоматизированной системы подачи изотопов водорода в ионный источник циклотрона У-400М и автоматизированного комплекса тритиевой мишени на сепараторе АКУЛИНА впервые были изучены резонансные уровни нейтронно-избыточных ядер ^4H и ^5H .

Автоматизированный стенд низкого давления ПРОМЕТЕЙ позволил впервые экспериментально подтвердить явление сверхпроницаемости для трития и измерить скорость мембранной откачки трития через ниобиевые и ванадиевые мембраны при различных давлениях и температурах.

Практическая ценность работы

Разработанный инструментальный пакет CRW-DAQ использовался для разработки программного обеспечения автоматизированных тритиевых комплексов установок ТРИТОН, АКУЛИНА и ПРОМЕТЕЙ. Кроме того, с его помощью автоматизированы исследования диффузии водорода в металлах в НИИФ СПбГУ и СПбГУТ, а также разработана система управления охлаждением и термостабилизацией спектрометра фотонов PHOS [20] в эксперименте ALICE [21] на Большом Адронном Коллайдере в ЦЕРН [22,23].

Автоматизированные тритиевые комплексы установок ТРИТОН и АКУЛИНА позволили провести исследования процессов мюонного катализа в смесях изотопов водорода в широком диапазоне температур, давлений и концентраций [24,25,26,27,28,29,30,31,32,33], получить нейтронно-избыточные ядра ^4H и ^5H и изучить их характеристики [34,35,36,37,38,39],

На автоматизированном стенде ПРОМЕТЕЙ проведены исследования явления сверхпроницаемости изотопов водорода через металлы, а также исследования явлений накопления и пропускания трития металлами и конструкционными материалами [40,41,42,43,44,45,46].

Опыт, накопленный при создании автоматизированных тритиевых комплексов установок ТРИТОН, АКУЛИНА и ПРОМЕТЕЙ обобщен в ряде публикаций [47,48,49,50,51], и может быть использован при решении задач

автоматизации других экспериментальных установок, в которых применяются изотопы водорода.

Личный вклад автора

Проанализированы требования к программному обеспечению для автоматизации тритиевых комплексов исследовательских установок и разработан программный пакет CRW-DAQ (около 140 тысяч строк кода) – инструментальная среда для разработки высоконадежного программного обеспечения для их автоматизации.

Решен ряд методических вопросов высоконадежного сбора данных и управления узлами комплекса подготовки газовой смеси (КПГС) и мишеней установки ТРИТОН, установки для подачи изотопов водорода в ионный источник циклотрона У-400М, комплекса жидко-тритиевой мишени установки АКУЛИНА и измерительных ячеек исследовательского стенда ПРОМЕТЕЙ.

Разработана основная часть алгоритмов сбора данных и управления для АСКУ тритиевых комплексов установок ТРИТОН, АКУЛИНА и ПРОМЕТЕЙ.

Автор участвовал в подготовке и проведении большинства физических экспериментов, выполненных на установках ТРИТОН и АКУЛИНА, а также в подготовке экспериментов на стенде ПРОМЕТЕЙ.

На защиту выносятся

1. Инструментальная программная среда (пакет CRW-DAQ), дающая качественно новые возможности для разработки высоконадежных, отказоустойчивых, радиационно-безопасных распределенных автоматизированных систем контроля и управления газовакуумными комплексами исследовательских установок, работающих с изотопами водорода, включая тритий.
2. Методические решения и программное обеспечение для автоматизации уникальных тритиевых мишенных комплексов на установках ТРИТОН и АКУЛИНА, созданных для изучения мюонного катализа ядерных реакций синтеза в смесях изотопов водорода и для изучения нейтронно-избыточных легких ядер, образующихся при взаимодействии пучка ионов трития с тритиевой мишенью, а также управления системой подачи изотопов водорода в ионный источник циклотрона для получения тритиевого пучка.

3. Система автоматизации управления, технологических и физических измерений на установке низкого давления ПРОМЕТЕЙ, предназначенной для изучения явлений сверхпроницаемости изотопов водорода, а также накопления и пропускания трития металлами и конструкционными материалами.

Апробация работы

Материалы, представленные в диссертации, докладывались на семинарах во ВНИИЭФ (г. Саров), ОИЯИ (г. Дубна); на 1, 2 и 3 международных семинарах "Взаимодействие изотопов водорода с конструкционными материалами (IHISM)", Саров'2001, Саров'2004, Санкт-Петербург'2007; международном Уральском семинаре "Радиационная физика металлов и сплавов", Снежинск'2003; на 1 и 7 международных конференциях "Мюонный катализ и связанные экзотические атомы. $\mu\text{SF-01}$, $\mu\text{SF-07}$ ", Япония, Шимода'2001, Дубна'2007; международной конференции по ядерной спектроскопии и структуре атомного ядра, Саров'2001; международных симпозиумах по экзотическим ядрам, Байкал'2001, Ладога'2004; международной конференции по тритиевой науке и технологиям, Германия, Баден-Баден'2004.

Публикации

Материалы, вошедшие в диссертацию, опубликованы в 20 печатных работах, в том числе 16 в реферируемых журналах: ВАНТ – 4, ЖЭТФ – 2, Материаловедение – 2, Приборы и техника эксперимента – 1, Известия РАН – 1, Ядерная Физика - 1, Nuclear Instruments and Methods – 2, Fusion Science and Technology – 2, Physics Letters B – 1.

Структура и объем диссертации

Диссертация состоит из введения, 3 глав, заключения и приложения. Общий объем 119 страниц. Диссертация содержит 57 рисунков, 6 таблиц и список использованных источников из 81 наименований.

Основное содержание работы

Во введении обоснована актуальность темы, определены цели работы, изложены научная новизна и значимость, отмечена научная и практическая ценность работы, сформулированы положения, выносимые на защиту.

Первая глава посвящена вопросам автоматизации комплекса подготовки газовой смеси (КПГС) и мишеней, созданных в РФЯЦ-ВНИИЭФ для установки ТРИТОН (ЛЯП ОИЯИ, г. Дубна), предназначенной для изучения процессов МК ядерных реакций в Н/D/T-смесях в диапазоне температур 20÷800 К, давлений до 160 МПа и активности трития в свободном состоянии до 10 кКи ($3,7 \cdot 10^{14}$ Бк).

Описан разработанный автором инструментальный программный пакет CRW-DAQ [19,47,48,49,50,51], предназначенный для разработки высоконадежных систем автоматизации газо-вакуумных комплексов исследовательских установок.

Описана структура многомашинной системы сбора данных и управления для комплекса подготовки газовой смеси и мишеней. Обсуждаются вопросы автоматизации при подготовке смесей ИВ, измерении их изотопного и молекулярного состава, измерения и стабилизации параметров смеси ИВ в мишенях различного типа, радиометрических измерений.

Приведены результаты большой серии экспериментов по изучению МК в D/T и D/D смесях, выполненных на установке ТРИТОН В 1997-2004 гг.

Вторая глава содержит краткое описание системы подачи изотопов водорода в ионный источник циклотрона У-400М, жидко-тритиевой мишени и комплекса её газового обеспечения на установке АКУЛИНА, предназначенной для экспериментов по изучению нуклонно-нестабильных легких ядер в ЛЯР ОИЯИ (г. Дубна).

Обсуждаются аппаратные и программные особенности управления системой подачи ИВ в ионный источник, управления комплексом мишени при ее

заполнении тритием, высокоточной стабилизации температуры мишени, а также радиационного контроля.

Приведены результаты использования этих автоматизированных систем при получении пучка ускоренных тритонов и подготовке криогенных мишеней (дейтериевой и тритиевой) в экспериментах по получению и изучению ядер ^4H , ^5H .

Третья глава содержит описание автоматизированного тритиевого комплекса низкого давления исследовательского стенда ПРОМЕТЕЙ, созданного в РФЯЦ-ВНИИЭФ. Основное внимание уделено вопросам управления экспериментами по изучению проникновения и накопления ИВ в металлах методом концентрационных импульсов (МКИ) и изучению явлений сверхпроницаемости ИВ через металлические мембраны. Описана система радиометрического контроля по тритию, предназначенная для обеспечения радиационной безопасности при эксплуатации стенда. Приведены данные о проведенных на автоматизированном стенде ПРОМЕТЕЙ экспериментах.

В заключении формулируются основные результаты, полученные в диссертационной работе.

В приложении приводятся дополнительные сведения о методических и программных решениях, которые применялись при реализации пакета CRW-DAQ и АСКУ тритиевых комплексов ТРИТОН, АКУЛИНА и ПРОМЕТЕЙ. Обсуждаются вопросы построения отказоустойчивых систем управления реального времени. Рассматриваются вопросы создания распределенных многомашинных систем управления с использованием разработанного в ЦЕРН протокола DIM и WEB-технологий. Обсуждаются детали реализации встроенных в пакет CRW-DAQ языков программирования. Приводятся дополнительные схемы и таблицы, иллюстрирующие устройство и работу АСКУ тритиевых комплексов ТРИТОН, АКУЛИНА и ПРОМЕТЕЙ.

1. Программное обеспечение для автоматизации измерений на установке высокого давления ТРИТОН

1.1. Установка ТРИТОН для исследования мюонного катализа ядерных реакций

Явление мюонного катализа ядерных реакций синтеза изотопов водорода [3] состоит в том, что мюон μ^- способен замещать электрон в молекулах ИВ, образуя мезомолекулы ($pp\mu$, $pd\mu$, $pt\mu$, $dd\mu$, $dt\mu$, $tt\mu$). Масса мюона в ~ 200 раз больше массы электрона и мезомолекула в ~ 200 раз меньше обычной молекулы, поэтому ядра d, t сближаются настолько, что могут туннелировать через кулоновский барьер, и инициировать реакцию синтеза, например:



Поскольку мюон не участвует в реакции, он может инициировать следующую реакцию синтеза, выступая катализатором ядерной реакции, подобно обычным катализаторам в химии. Процесс МК ограничен временем жизни мюона и химическими факторами: скоростью образования мезомолекул и «прилипанием» мюона к радиогенному гелию при образовании мезоатома $\alpha\mu$, которое действует подобно «отравлению» катализатора в химии. Важной особенностью МК является зависимость течения ядерной реакции синтеза от макроскопических (температура T, плотность ρ , давление P) и химических (состав Н/Д/Т смеси) условий, влияющих на образование мезомолекул.

Для исследования мюонного катализа (МК) ядерных реакций синтеза в Н/Д/Т-смесях в диапазоне температур 20÷800К, давлений до 160 МПа и активности трития в свободном состоянии до 10 кКи ($3,7 \cdot 10^{14}$ Бк) на мезонном пучке фазотрона ЛЯП ОИЯИ в 1996 году была создана установка ТРИТОН. Эта работа проводилась в рамках программы научно-технического сотрудничества РФЯЦ-ВНИИЭФ и ОИЯИ (Дубна).

Установка ТРИТОН состоит из ядерно-физического и тритиевого комплексов. Ядерно-физический комплекс (Рис.1,а) создан специалистами ЛЯП ОИЯИ для регистрации частиц от реакций d-t синтеза и распада мюонов [25]. Сцинтилляционные счетчики 1, 2, 3 и пропорциональный счетчик 4 регистрируют

прохождение мюона в мишень. Детекторы полного поглощения ND1, ND2 регистрируют нейтроны, возникающие в реакции d-t синтеза. Пропорциональный счетчик 5 и сцинтилляторы 1-е, 2-е регистрируют электроны от распада мюонов.

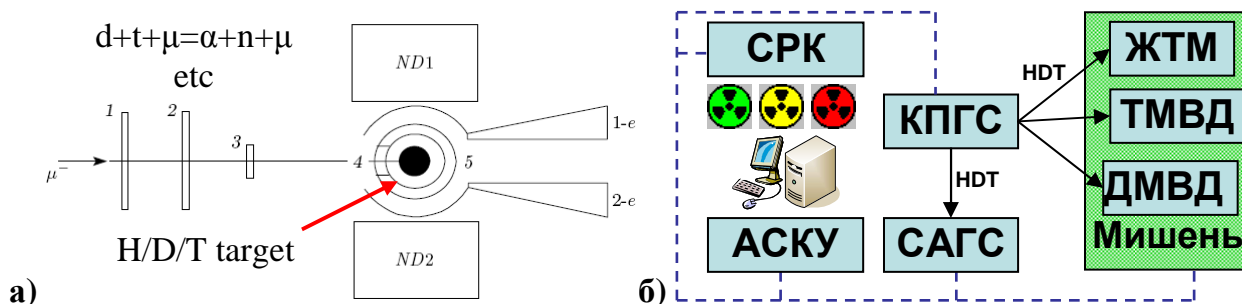


Рис.1. Ядерно-физический (а) и тритиевый (б) комплекс установки ТРИТОН.

В задачу специалистов РФЯЦ ВНИИЭФ входило создание радиационно-безопасного тритиевого комплекса, обеспечивающего безаварийную работу с нужными количествами трития в условиях неспециализированной лаборатории [7].

1.2. Структура тритиевого комплекса установки ТРИТОН

Тритиевый комплекс (Рис.1,б) служит для подготовки смеси ИВ требуемого изотопного состава, заполнения мишени, контроля и стабилизации температуры и давления в мишени, контроля радиационной обстановки. Он включает комплекс подготовки газовой смеси (КПГС) [6], криогенную установку [52] и сменные мишени: жидко-тритиевую мишень (ЖТМ) [53], тритиевую мишень высокого давления (ТМВД) [54], дейтериевую мишень высокого давления (ДМВД) [55]. В него также входят: система анализа состава газовой смеси (САГС) [56,57], система радиационного контроля по тритию (СРК) [58] и АСКУ [8].

Конструктивно КПГС (Рис.2) состоит из 10 связанных между собой подсистем (Рис.3). Герметичный бокс, где находятся основные элементы КПГС, обеспечивает подготовку смеси заданного состава и её подачу к мишени. Вакуумный пульт выполняет вакуумирование рабочих газовых коммуникаций, утилизацию основного количества тритийсодержащей газовой смеси и молекулярный анализ газовой смеси. Ресивер нужен для сброса «хвостов» газовой смеси с выхлопов форвакуумных насосов. Установка газовой очистки нужна для очистки газовых смесей от следов трития и его соединений, находящихся в герметичных объемах (боксе, ресивере и т.п.). Установка раннего предупреждения предотвращает выброс в атмосферу газовых смесей, содержащих тритий выше

установленных норм. Сборка насосов вакуумирует ресивер, газовые линии, герметичные технологические объемы и т.п. Баллонный пост обеспечивает подачу диффузионно-чистого протия и дейтерия к КПГС, заполнение герметичных технологических объемов инертным газом и питание газового радиохроматографа газом-носителем. Система радиометрического контроля следит за состоянием элементов КПГС и радиационной обстановкой в рабочей зоне. Электрический пульт дает ручное управление и контроль КПГС и мишени.

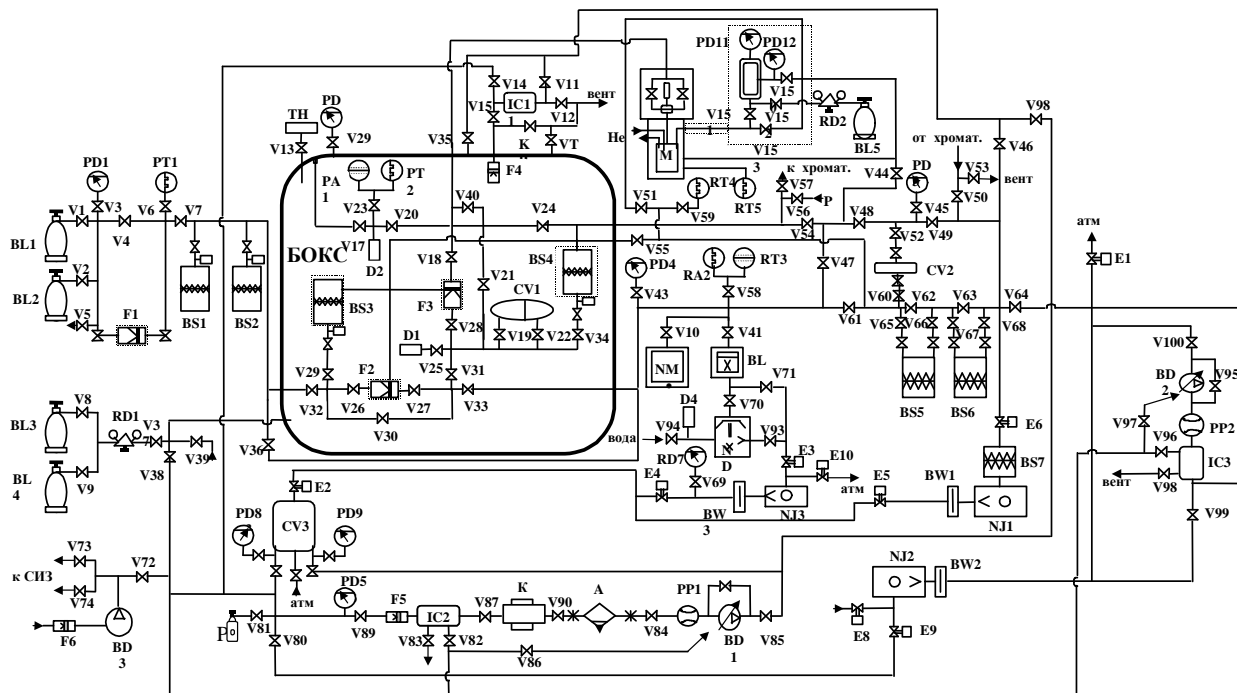


Рис.2. Схема газовых коммуникаций КПГС установки ТРИТОН [6].

BS1-BS2 - генераторы протия и дейтерия; BS3 - генератор трития; BS4 - смесевой генератор; BS5, BS6 - урановые ловушки; BS7 - ловушка адсорбционная; PA - преобразователь ионизационный ПМИ-2; PT - преобразователь термопарный; CV1, CV2 - емкости; CV3 - ресивер; CV4 - емкость водородная; D1, D2 - датчик давления "Сапфир-22-Ех-М"; М - тритиевая мишень высокого давления; ТН - тягонапорометр; IC - камера ионизационная; NM - насос магнитоэлектрический; ND - насос диффузионный; NJ - насос форвакуумный; PP - ротаметр; P - пробоотборник; Т - выключатель гидравлический; К - конвертор; А - адсорбер; BD - воздуходувка; BW - фильтр маслоотбойный; F1-F3 - фильтры диффузионные; BL - баллоны с газами; F4-F6 - фильтры аэрозольные; V - вентили; E - клапаны электромагнитные; PD - стрелочные манометры, вакууметры и моновакууметры; KP - клапан предохранительный; RD - газовые редукторы; СИЗ - средства индивидуальной защиты.

Для такой сложной установки, с учетом ее радиационной опасности, необходима система автоматизированного управления. Управление комплексом при этом значительно упрощается, становится более надежным и предсказуемым, снижается влияние на устойчивость работы установки "человеческого фактора" (замедленная реакция, ошибки по невнимательности), что особенно важно при круглосуточной, непрерывной работе на установке во время сеанса измерений.

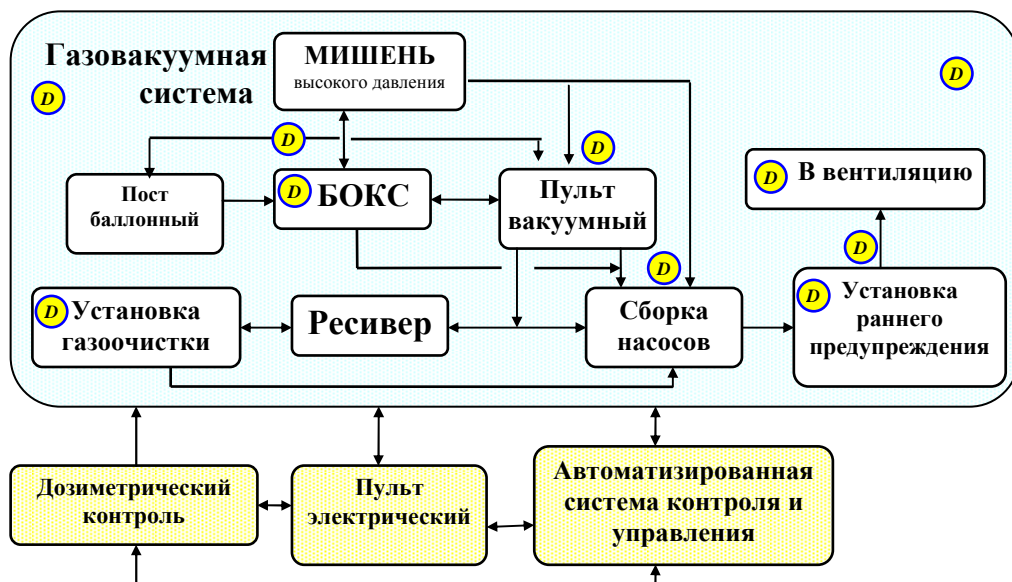


Рис.3. Структурная схема КПГС. D – точки дозиметрического контроля.

На АСКУ возложен ряд задач – это обеспечение технологических измерений (измерения температур, давлений и вакуума) и управление узлами комплекса (вакуумными насосами, вентилями и клапанами), регулирование температуры источников и фильтров; измерение изотопного и молекулярного состава газовой смеси для наполнения мишени, измерение и поддержание параметров газовой смеси в мишени в ходе эксперимента. Она также служит для блокировки исполнительных устройств (при возникновении аварийно-опасных ситуаций), аварийного оповещения в случае радиационной опасности, ведения протокола эксперимента, визуализации состояния комплекса в реальном времени и первичной математической обработки данных.

1.3. Аппаратурное и программное обеспечение АСКУ тритиевого комплекса

АСКУ конструктивно выполнена в виде трех подсистем в сети Ethernet, каждая базируется на своем ПК (Рис.4). РС_1 обеспечивает работу КПГС [6] и СРК [58], РС_2 – работу с мишенью (ЖТМ [53], ТМВД [54] и ДМВД [55]), а РС_3 – контроль САГС [56,57]. Такое разделение учитывает необходимость наблюдения большого количества физических и технологических параметров одновременно, а также совместной работы группы операторов со всеми подсистемами комплекса. При этом КПГС и СРК используются непрерывно, от подготовки сеанса до его завершения, подсистема мишени - при подготовке, заполнении и эксплуатации мишени, а САГС - периодически для изотопного анализа проб газовой смеси.

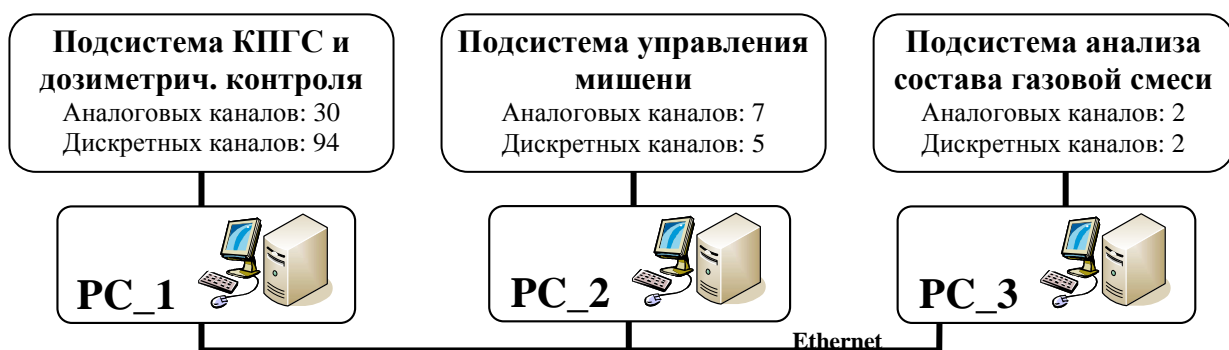


Рис.4. Общая структура АСКУ тритиевого комплекса.

АСКУ построена в виде распределенной сети интеллектуальных модулей, связанных с управляющим компьютером по стандарту RS-232 и RS-485. В ней используется 23 измерительных устройств, включая 17 модулей серии I-7000, 2 контроллера TPG-256, 3 интеллектуальных контроллера КРО и карту цифрового ввода-вывода DIO-144. Суммарно система имеет 140 измерительных каналов, включая 101 цифровых и 39 аналоговых, при этом управляющими являются 30 каналов (Таблица 1).

Таблица 1. Список измерительных каналов АСКУ ТРИТОН.

Измерение и управление	Каналов	Тип канала
ТРИТОН - КПГС		
Термопарные измерения температуры нагревателей источников	8	аналоговый
Управление нагревателями источников	8	дискретный
Контроль обрыва спиралей нагревателей источников	8	дискретный
Управление релейной блокировкой нагревателей источников	8	дискретный
Измерение низких давлений датчиками вакуума фирмы Balzers	2	RS-232
Контроль вакуума лампами ПМТ-4	4	аналоговый
Контроль тока ламп ПМТ-4	4	аналоговый
Измерение высоких давлений тензометрическими датчиками	6	аналоговый
Измерение давлений датчиками типа «Сапфир»	2	аналоговый
Контроль состояний датчиков давления воды	4	дискретный
Контроль состояний ручных вентилях с датчиками положения	48	дискретный
Управление электромагнитными клапанами	9	дискретный
Контроль состояний электромагнитных клапанов	9	дискретный
Контроль объемной активности трития в рабочем помещении	4	RS-485
ТРИТОН - мишень		
Термопарные измерения температуры нагревателя мишени	1	аналоговый
Управление нагревателем мишени	1	дискретный

Контроль обрыва спирали нагревателя мишени	1	дискретный
Управление релейной блокировкой нагревателей мишени	1	дискретный
Контроль состояний ручных вентилях с датчиками положения	2	дискретный
Измерение температуры мишени криогенными термодиодами	2	аналоговый
Стабилизация температуры мишени	1	аналоговый
Измерение давления датчиком типа «Сапфир»	1	аналоговый
Измерение высокого давления тензометрическими датчиками	2	аналоговый
ТРИТОН - радиохроматограф		
Измерение сигнала датчика по теплопроводности	1	аналоговый
Управление диапазоном ЭМУ-2	2	дискретный
Измерение тока ионизационной камеры	1	аналоговый

Для аналогового и дискретного контроля и управления узлами КПГС используются модули серии I-7000 фирмы ICP DAS, выбор которых обоснован в разделе 1.4.2.7. Для контроля вакуума служат датчики фирмы Balzers, подключенные к контроллерам TPG-256, оснащенным интерфейсом RS-232. Для контроля состояний вентилях и клапанов используется карта дискретного ввода/вывода DIO-144 на шине ISA. Для контроля радиационной обстановки и анализа газовой смеси используется специальное оборудование (разделы 1.5, 1.7).

Программное обеспечение АСКУ комплекса, являющегося распределенной многомашинной системой, разработано базе описанного ниже инструментального пакета CRW-DAQ [19]. Ядро пакета запускается на всех компьютерах АСКУ при старте ОС. Конкретная программа работы на каждом компьютере определяется загружаемым файлом конфигурации на языке DAQ Config (раздел 1.4.2.5). Конфигурационные файлы ссылаются на прикладные программы, содержащие алгоритмы управления на языке DAQ Pascal, калибровки, мнемосхемы и другие ресурсы.

1.4. Инструментальный программный пакет для автоматизации физических экспериментов

Современные промышленные системы автоматизации обычно строятся на базе серийной модульной цифровой измерительной аппаратуры и промышленных компьютеров [59], с использованием специализированных программных инструментов (SCADA), позволяющих скрыть от разработчика прикладных

программ сложность системного программирования, предоставив ему простой программный интерфейс для создания прикладных алгоритмов управления. Для решения стандартных задач автоматизации в промышленности обычно используются коммерческие пакеты SCADA (Lab View, Trace Mode, WinCC и т.д.), коды которых разработчику не предоставляются.

Для автоматизации исследовательских установок и физических измерений использование таких пакетов затруднено по ряду причин. Так, например, в тритиевых комплексах исследовательских установок применяется специальное измерительное оборудование - аппаратура для вакуумных измерений, масс-спектрометрии, газовой радиохроматографии и изотопного анализа Н/Д/Т смесей; техника для измерений в области криогенных температур; устройства для контроля объемной активности трития в технологических объемах и воздушной среде. При использовании закрытых коммерческих пакетов возникают проблемы с драйверами нестандартных устройств. Кроме того, тритиевые установки являются опасными объектами и требуют полной предсказуемости поведения системы управления, которую трудно гарантировать при использовании закрытых пакетов.

По этим причинам открытое ПО представляется более привлекательным для поставленных задач. Открытый исходный код дает полную информацию о деталях работы системы управления, позволяя сделать её поведение более предсказуемым и надежным. Другое важное достоинство открытого ПО – полный контроль над развитием проекта и независимость от конкретного производителя. Судьба таких крупных фирм, как DEC, Sun, Borland, Ericson и других показывает, что «фирменность» еще не является гарантией устойчивого развития проектов ПО.

По указанным причинам с 1996 года во ВНИИЭФ была начата разработка собственного инструментального программного пакета, названного CRW-DAQ, предназначенного для решения двух основных задач - разработки систем сбора данных и управления, а также для первичной обработки измеренных экспериментальных данных [47,48,49,50,51]. В 2006 г. пакет был официально зарегистрирован в государственном реестре Российской Федерации [19].

Название пакета (CuRves in Windows for Data Acquisition) отражает идею создания графически ориентированной среды для физиков-экспериментаторов, в которой работа происходит в основном с графиками данных (кривыми в окнах), а

не с таблицами, как в ряде других пакетов. Пакет разрабатывался на языке Object Pascal с помощью компилятора Delphi 5 [60]. Пакет работает в операционных системах Windows-NT/2000/XP/Vista/7 и использует возможности вытесняющей приоритетной многозадачности, предоставляемые ядром Windows-NT. Для АСКУ тритиевых комплексов достаточен программный опрос с периодом $1 \div 10$ мс, которого позволяет добиться потоковая модель Windows-NT, так как основные исполнительные устройства (клапаны, насосы, нагреватели) требуют времени реакции порядка 100 мс, см. также Приложение 1.

Учитывая большое число и сложность SCADA систем, их сравнение могло бы стать темой отдельного исследования. Таблица 2 содержит лишь краткое сравнение пакета CRW-DAQ с другими. Достоинства отмечены зеленым цветом, а недостатки - желтым. Ряд различий нельзя однозначно отнести к достоинствам или недостаткам. Так, текстовая среда разработки является и недостатком (её труднее осваивать), и достоинством (она дает возможность автоматизации поиска\замены переменных, сравнения версий программ, анализа кода и т.д.).

Таблица 2. Сравнение пакета CRW-DAQ с другими.

Свойство	CRW-DAQ	LabView, WinCC и т.д.
Ориентация	Исследования	Промышленность
Тип лицензии	Открытая, свободная	Закрытая, коммерческая
Поддержка и развитие	Силами лаборатории с 1998	Фирменная
Документация	Есть, достаточная	Есть, обширная
Книги, журналы	Нет	Есть
Примеры, шаблоны	Есть, много	Есть
Исходный код, Открытость реализации	Открытый, Принцип «ЗНАЮ КАК»	Закрытый, Принцип «ЧЕРНЫЙ ЯЩИК»
Зависимость от фирмы	Никакой	Полная
Лицензионные ограничения	Нет	Есть, зависит от продукта
Среда разработки	Текстовая	Визуальная
Синхронизация версий ПО	Есть (на базе windiff)	Нет (двоичные коды)
Система калибровки	Есть, развитая	Готовой нет
Самодиагностика, мониторинг, журналирование	Есть, развитая	Готовой нет
Многопоточность	Есть, всегда	Потенциально
Компиляция «на лету»	Есть, не прерывая измерений	Нет, требуется перезагрузка
Отказоустойчивость	Есть, виртуальная машина	Потенциально
Драйверы серийных устройств	Есть, достаточно	Есть, очень много
Драйверы спец. устройств для третичных комплексов	Есть, легко добавить новые	Нет, но можно добавить
Разработка драйверов спец.устр.	Есть, встроенная	Есть, требует доп.инструментов

В таблице нет сравнения по ряду численных ограничений (максимальное число каналов, частота опроса и т.д.), т.к. сравнение параметров закрытого коммерческого ПО с открытым ПО, где такие ограничения легко снимаются перекомпиляцией программы с другими параметрами, лишено всякого смысла.

Резюмируя, можно сказать, что разработанный пакет, не претендуя на конкуренцию с промышленными гигантами, в то же время хорошо подходит для своего круга задач (автоматизация исследовательских установок). Успешную автоматизацию и эксплуатацию установок ТРИТОН, АКУЛИНА, ПРОМЕТЕЙ и других можно рассматривать как подтверждение этого.

1.4.1. Общая структура пакета

Программное обеспечение АСКУ в пакете CRW-DAQ делится на два класса: базовое и прикладное (Рис.5). Это разделение облегчает техническую поддержку АСКУ и позволяет повысить надежность прикладных программ.

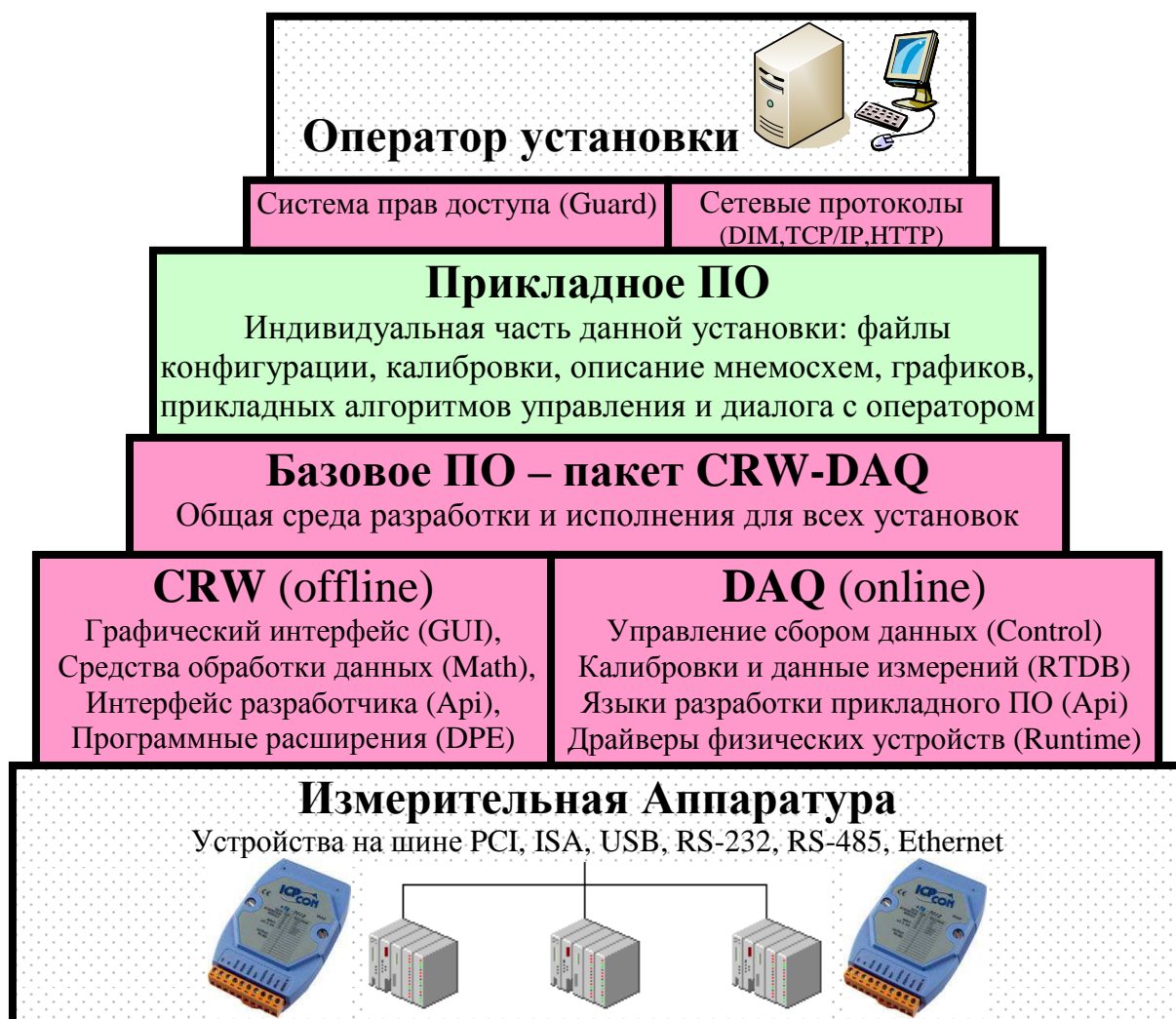


Рис.5. Структура программного обеспечения АСКУ в пакете CRW-DAQ.

Базовое ПО – это сам пакет CRW-DAQ, обеспечивающий единую для всех АСКУ совмещенную графическую среду разработки и исполнения, а также единый программный интерфейс и встроенные языки программирования. Базовое ПО пакета можно условно разделить на подсистемы CRW и DAQ (Рис.5).

Подсистема CRW не связана прямо со сбором данных и управлением. Она содержит средства и библиотеки общего назначения для визуализации и графического интерфейса (GUI), математической обработки данных (Math), включает программный интерфейс (API) для динамических расширений (DPE), систему ограничения прав доступа (Guard), библиотеки поддержки драйверов устройств и сетевых протоколов (DIM, TCP/IP, HTTP), инструменты разработчика, справочную систему и т.д. Эту часть пакета можно использовать автономно, например, для наблюдения и offline обработки измеренных данных.

DAQ содержит средства для разработки, запуска (Control) и исполнения (Runtime) кода прикладных АСКУ, ПО для которых создается с помощью встроенных ЯВУ (DAQ Config, DAQ Pascal) и программных интерфейсов (DAQ API). Ядром исполнительной системы DAQ служит база данных реального времени (RTDB), где хранятся все переменные состояния АСКУ, нужные для её работы.

Прикладное ПО реализует специфику конкретной измерительной системы и содержит средства, индивидуальные для каждой АСКУ. Оно располагается в отдельном каталоге, не входящем в базовую инсталляцию пакета CRW-DAQ, и состоит из конфигурационных файлов, задающих параметры и структуры данных, и прикладных программ, задающих алгоритмы управления и написанных на встроенных в пакет языках прикладного программирования.

Таким образом, пакет CRW-DAQ является самодостаточной инструментальной средой и не требует для создания ПО АСКУ дополнительных программ (редакторов и компиляторов), кроме операционной системы Windows и драйверов устройств, поставляемых вместе с серийным оборудованием. При этом пакет CRW-DAQ служит средой разработки и исполнения прикладного ПО АСКУ.

1.4.2. Компоненты пакета

1.4.2.1. Компоненты для организации графического интерфейса пользователя

Для визуализации измеряемых данных в реальном времени и организации диалога с пользователем в пакете CRW-DAQ (Рис.5) разработан компонент CRW GUI (Graphics User Interface), содержащий многооконный графический интерфейс, основанный на принципе MDI (Multi Document Interface) и реализованный с помощью библиотеки VCL (Visual Component Library). Основными элементами GUI являются окна – редакторы, консольные окна, двух и трехмерные графики, мнемосхемы, таблицы и диалоговые окна. Принцип MDI означает, что главное окно программы (Рис.6) является контейнером, внутри которого содержатся дочерние окна, не выходящие за пределы главного окна.

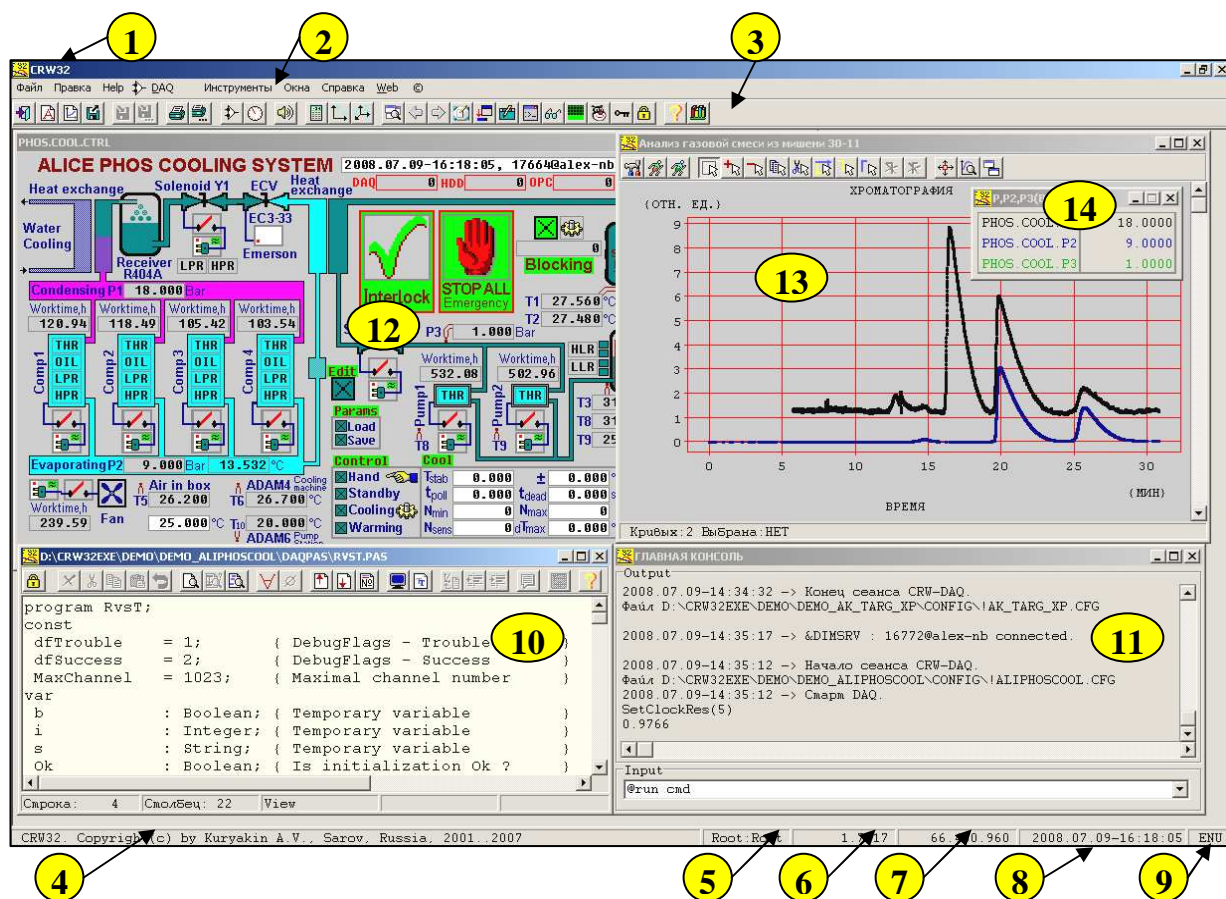


Рис.6. Внешний вид графического интерфейса главного окна пакета CRW-DAQ.

Заголовок (1); панель меню (2); панель инструментов (3); строка статуса (4); индикатор прав доступа (5); индикатор реального времени (6); индикатор памяти (7); индикатор календарного времени (8); индикатор языка (9); холст (основное поле) для дочерних MDI окон, включая редакторы текста (10), консольные окна (11), мнемосхемы (12), графики (13) и таблицы (14).

Для редактирования текстов прикладных программ и конфигураций реализованы текстовые окна (Рис.6.10) нескольких типов: простой текст, редактор

программ DAQ Pascal, редактор программ Object Pascal и т.д. Кроме обычных команд текстового редактора, эти окна содержат ряд зависящих от типа окна специальных функций. Например, команда компиляции вызывает внутренний (для программ DAQ Pascal) или внешний (для программ Object Pascal) компилятор для преобразования текста программы в исполняемый код, команда запуска выполняет откомпилированный исполняемый код, а команда справки вызывает контекстную справку с описанием используемого компилятора.

Для отображения текстовой информации в реальном времени разработаны консольные окна (Рис.6.11), специально оптимизированные для работы в многопоточном режиме с помощью FIFO буферов для ввода/вывода пользовательской информации на экран. Поэтому процедуры консольного ввода-вывода (readln, writeln) в прикладных программах не мешают работе алгоритмов реального времени. Это делает консольное окно универсальным средством отображения, пригодным как для диагностики и отладки, так и для интерфейса с пользователем. Команды копирования/вставки позволяют переносить консольный вывод реального времени в текстовые окна для последующего анализа.

Для отображения текущих численных значений измеряемых величин в реальном времени разработаны окна таблиц (Рис.6.14), также оптимизированные для работы в многопоточном режиме.

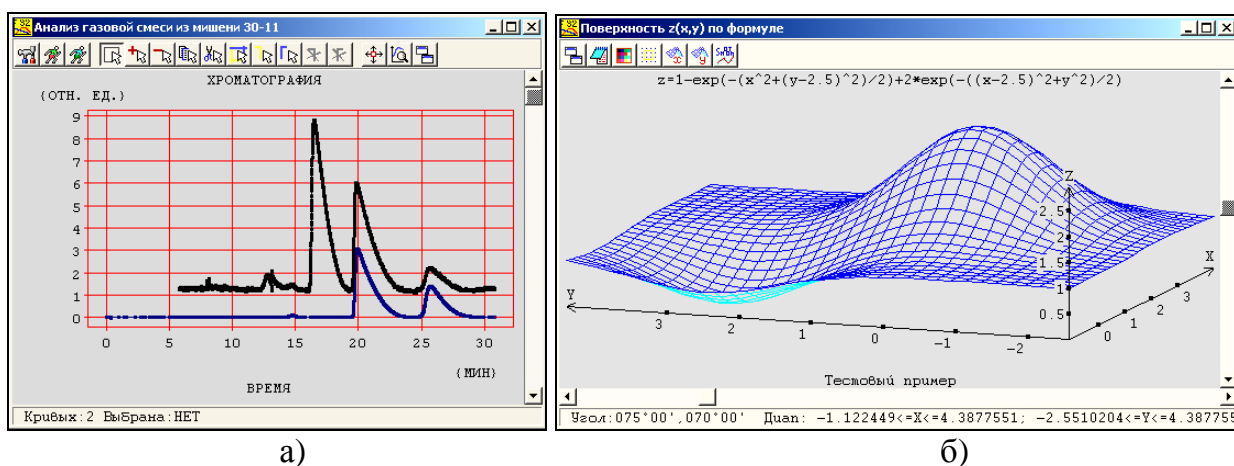


Рис.7. Окна для отображения графиков кривых (а) и поверхностей (б).

Для отображения математических зависимостей в виде графиков в реальном масштабе времени разработаны окна кривых (Рис.6.13, Рис.7,а). Математические зависимости $y(x)$ представлены в пакете CRW-DAQ своим дискретным аналогом – кривой, т.е. упорядоченным массивом точек (x_i, y_i) , $i=1...N$. Чаще всего абсцисса x

соответствует времени, а ордината y – некой измеряемой величине. Массив абсцисс обычно упорядочен по возрастанию времени, т.е. $x_{i+1} \geq x_i$. Окно кривых содержит коллекцию (список произвольной длины) кривых, одна из которых может быть выбрана для отображения или проведения над ней операций обработки данных.

Окна кривых играют важную роль в пакете CRW-DAQ, так как к ним привязаны команды математической обработки данных. Файловые команды позволяют создавать окна, загружать или сохранять данные в файле. Командами редактирования можно удалять, вырезать, копировать или вставлять кривые. Команды курсора мыши позволяют выбирать мышью видимый фрагмент окна, увеличивать и уменьшать его, копировать или переносить кривые из окна в окно методом «Drag & Drop», выделять прямоугольную область интереса ROI (Region Of Interest). Командами масштабирования можно увеличивать, уменьшать и сдвигать пределы графика, а также задавать их вручную или автоматически. Команды настройки служат для редактирования цвета, толщины и стиля кривой, сопроводительного текста кривой, внешнего вида окна. Команды выбора позволяют указать кривые для последующих операций. Команда клонирования создает копию окна вместе со всеми данными. Команда табулирования генерирует текстовую таблицу выбранной кривой для экспорта данных в другие пакеты. Команда печати не только печатает на принтере, но и позволяет копировать окно в буфер обмена, как в виде изображения, так и в виде текстовой таблицы. Это позволяет легко переносить данные из окна в стандартные пакеты (Word, Excel) путем «печати» в буфер обмена и вставки в электронную таблицу. Команды вызова математических утилит позволяют обрабатывать измеренные данные с помощью динамически загружаемых расширений DPE (dynamic plug-in extension) в виде DLL, написанных на языке Object Pascal, или при помощи макросов на языке DAQ Script. Эти команды дают доступ к десяткам уже готовых утилит обработки данных (математического анализа, фильтрации, сглаживания), а также к средствам разработки новых утилит для вновь возникающих задач (см. раздел 1.4.2.4).

Для отображения графиков функций двух аргументов $z(x,y)$ в виде аксонометрической проекции их «каркаса» на равномерной прямоугольной сетке (x_i, y_j) созданы окна поверхностей (Рис.7,б). Исходные данные $z(x,y)$ могут быть заданы на любой сетке в виде массива (x_i, y_j, z_i) , т.к. для вычисления функции

используется линейная 3-точечная интерполяция и триангуляция Делоне [61]. Окна поверхностей применяются для offline обработки данных после измерений.

Для эффективной работы АСКУ должна иметь интуитивно понятный интерфейс пользователя. Поэтому измерительным и исполнительным устройствам должны соответствовать визуальные элементы в виде общепринятых и очевидных для специалистов данной области (газовые установки, физика ИВ) изображений.

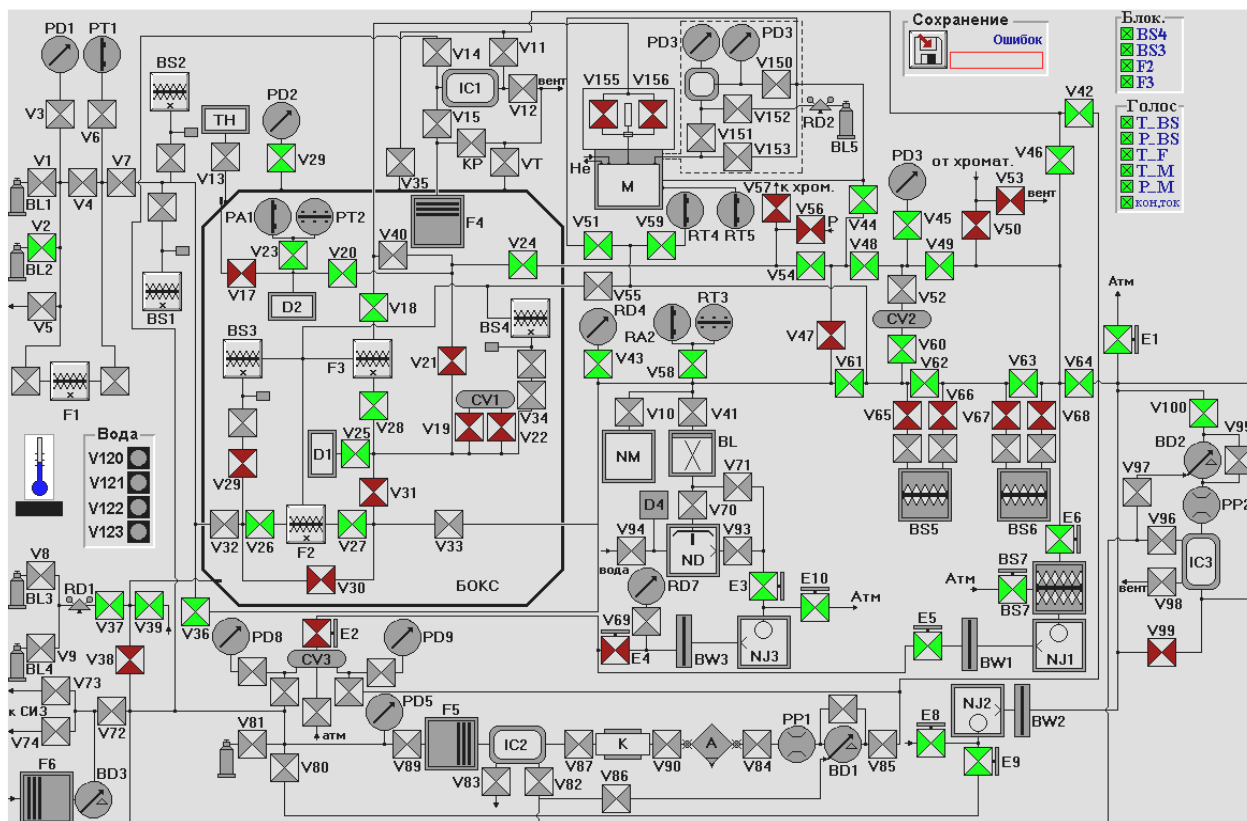


Рис.8. Главная мнемосхема КПГС установки ТРИТОН.

Для создания таких интерфейсов разработаны окна мнемосхем (Рис.6.12). Они содержат постоянное основное поле в виде растрового изображения, на котором помещены сенсоры – чувствительные к действиям пользователя элементы мнемосхемы, представленные растровыми изображениями, которые могут меняться в зависимости от состояния связанных с ними элементами данных (тегами или кривыми). Для газовых установок в качестве основного поля мнемосхемы обычно берется изображение схемы газовых коммуникаций (Рис.8). В качестве сенсоров выступают условные изображения физических устройств: клапанов, вентилей, нагревательных элементов, насосов, атомизаторов, датчиков воды, ионизационных камер и т.д. На больших установках мнемосхем может быть много – главная мнемосхема и несколько вспомогательных. Главная мнемосхема

позволяет быстро оценить состояние установки в целом и вызвать мнемосхемы, позволяющие работать с отдельными подсистемами более подробно.



Рис.9. Модель взаимодействия потоков визуализации и управления в пакете CRW-DAQ.

При реализации графического интерфейса (Рис.9) целью ставилась в первую очередь быстрая реакция программы на события в системе управления в реальном времени. Поэтому все окна (консоли, мнемосхемы, таблицы, графики кривых) оптимизированы для работы в многопоточном режиме и не влияют на работу высокоприоритетных измерительных потоков. Для этого поток визуализации и измерительные потоки изолированы друг от друга и взаимодействуют не прямо, а только через промежуточный компонент – тщательно оптимизированную и хорошо защищенную базу данных реального времени RTDB, содержащую общие для потоков данные (теги, кривые, FIFO буферы). Сбор данных и управление идет в измерительных потоках с высоким приоритетом и периодом опроса $\sim 1 \div 10$ мс, данные записываются в RTDB. Для обновления изображения применяется низкоприоритетный поток с таймером ~ 55 мс, выбранным исходя из типичного времени реакции человека. Это снижает взаимное влияние потоков до минимума.

Главным достоинством такой организации GUI является независимость потоков друг от друга, позволяющая измерительным потокам с высоким приоритетом вытеснять низкоприоритетные потоки визуализации, когда потребуется, что необходимо для задач управления в реальном времени.

1.4.2.2. Компоненты для математического анализа данных

Одним из необходимых инструментов физика-экспериментатора являются средства цифровой фильтрации и сглаживания, так как измеренные данные обычно дискретны по времени и амплитуде, зашумлены и искажены помехами.

Для online обработки измеряемых данных в реальном времени, а также для offline обработки измеренных данных в интерактивном режиме в пакете разработан компонент CRW Math (Рис.5). Он содержит математические библиотеки, программные интерфейсы, а также графически-ориентированные средства для

математического анализа измеренных данных, фильтрации, сглаживания, импорта и экспорта данных в стандартные математические пакеты (Excel, Origin и т.д.).

Пусть имеется массив измеренных данных $\{x_i, y_i\}$, $i=1..N$, дающий дискретное приближение к неизвестной зависимости $y_{(x)}$. Измеренный сигнал обычно содержит помехи (электромагнитные наводки, электронный шум, квантование по амплитуде и т.д.), поэтому его можно представить в виде

$$y_i = y_{(x_i)} + e_i, \quad i = 1..N \quad (2)$$

где e_i - обобщенная ошибка, включающая все виды помех.

Восстановление сигнала (2) с помехами является некорректной задачей, разрешимой лишь при наличии дополнительной априорной информации о нем. Это может быть параметрическая модель, когда искомая кривая $y_{(x)}$ подгоняется модельной функцией $f_{(x,p)}$ с обобщенным параметром p , или общие свойства сигнала: непрерывность, монотонность и т.д. Соответственно методы сглаживания делятся на параметрические, требующие аналитической модели зависимости $y_{(x)}$, и непараметрические, использующие лишь качественную информацию о сигнале.

Задачу сглаживания можно сформулировать как процедуру получения устойчивой по отношению к помехам оценки $y_{(x,p)}^*$ значения $y_{(x)}$ в произвольной точке x с помощью некоторого метода с обобщенным параметром p , которая с одной стороны удовлетворяет априорной информации о сигнале, а с другой наиболее близка к измеренным данным. В частности, делая оценки в узловых точках x_i , получаем сглаженный массив $\{x_i, y_{i,p}^*\}$. В качестве критерия близости к измеренным данным берется, например, невязка

$$\delta(y, y_p^*) = \sum_{i=1}^N (y_{i,p}^* - y_i)^2 \cdot w_i \quad (3)$$

т.е. взвешенная сумма квадратов отклонений с некоторыми весами $w_i > 0$. Таким образом, при сглаживании всегда присутствует компромисс между априорной (теоретической) и апостериорной (экспериментальной) информацией о сигнале.

Задачи управления связаны с выработкой сигналов управления в петле обратной связи на основе оценки $y_{(x,p)}^*$. Они требуют обработки в режиме online, в реальном времени, и это накладывает на методы обработки особые требования.

Своевременность оценки требует, чтобы она получалась за предсказуемое, ограниченное время, чтобы АСКУ успела вовремя выдать управляющий сигнал. Некоторые итеративные методы не имеют предсказуемого времени расчета.

Однозначность оценки требует, чтобы метод давал единственную оценку, иначе система управления не сможет выработать реакцию. Например, метод наименьших квадратов (МНК), может давать несколько решений (локальных минимумов), поэтому может оказаться непригодным для задач управления.

Надежность оценки требует гарантии получения разумного значения $y_{(x,p)}^*$ при любых, даже сильно поврежденных входных данных, т.к. АСКУ в любом случае обязана выработать какой-то сигнал управления. Некоторые итеративные методы ненадежны, т.к. итерации могут не сходиться.

Оценка должна выполняться на основании выборки $\{x_i, y_i\}$, которая может быть ограничена причинно и исторически. Причинное ограничение связано с тем, что при получении оценки $y_{(x,p)}^*$ в момент x доступны только прошлые по времени данные, а не вся зависимость, как в offline режиме. Будущие измерения еще не выполнены, поэтому выборка $\{x_i, y_i\}$ может содержать только данные о прошлом:

$$\{x_i, y_i\} : x_i \leq x \quad (4)$$

Историческое ограничение возникает, если из-за недостатка времени или памяти для обработки доступен только небольшой объем данных не более n точек:

$$\{x_i, y_i\} : i \leq n \quad (5)$$

Выборки $\{x_i, y_i\}$ в задачах управления часто расположены на случайной сетке, когда условие постоянства шага по времени не соблюдается:

$$x_{i+1} - x_i \neq const \quad (6)$$

Так, неравномерность поступления данных наблюдается в распределенных АСКУ, где измерения идут на нескольких независимых компьютерах или контроллерах, а измеренные данные поступают по сети Ethernet, RS-232, RS-485, CAN и т.д. Неравномерность сетки ограничивает применение таких методов, как цифровые фильтры или wavelet анализ, рассчитанных на постоянную сетку по времени.

В то же время выборки $\{x_i, y_i\}$ в измерительных задачах практически всегда монотонны по времени, т.е. содержат упорядоченные по абсциссе данные:

$$x_i \leq x_{i+1} \quad (7)$$

так как они получены в реальном времени, в ряде последовательных измерений. Это позволяет применять процедуры быстрого двоичного поиска в массивах данных, что резко ускоряет вычисления.

Указанные особенности обработки данных в реальном времени сильно повлияли на выбор алгоритмов, включенных в пакет CRW-DAQ.

Для использования параметрических методов сглаживания в пакете CRW-DAQ реализована библиотека функций минимизации по МНК для линейных и нелинейных моделей. Он относится к нелокальным, нелинейным параметрическим методам, может работать с данными на случайной сетке. В качестве оценки $y_{(x)}^*$ принимается значение $y_{(x)}^* = f_{(x,p^*)}$, где искомым параметр p^* находится процедурой минимизации функции невязки в виде взвешенной суммы квадратов:

$$p^* = \arg \min \left(\sum_{i=1}^N (y_i - f_{(x_i,p)})^2 \cdot w_i \right) \quad (8)$$

В качестве весов берутся обратные величины $w_i = \rho_i^{-1}$ дисперсии ρ_i измерений в i -й точке или единичные веса $w_i = 1$, если дисперсии неизвестны. Библиотека МНК доступна как набор функций CRW API на языке Object Pascal, а также в виде готовых инструментов с графическим интерфейсом.

В библиотеке МНК реализовано 8 алгоритмов [66]: симплексный метод Нелдера-Мида (Nelder-Mead), метод сопряженных градиентов Флетчера-Ривза (Fletcher-Reeves), квази-Ньютоновские методы Давидона, Давидона-Флетчера-Пауэлла (Davidon, Fletcher, Powell), Бройдена (Broyden), Пирсона (Pearson), Заутендайка (Zoutendijk), Стюарта (Steward). Алгоритмы Нелдера-Мида и Стюарта не требуют вычисления аналитического градиента, что на практике является достоинством, но работают надежно лишь при небольшой размерности (≤ 5 параметров). Другие методы требуют вычисления градиента целевой функции и обладают квадратичной сходимостью, т.е. требуют конечного числа итераций, если целевая функция является квадратичной формой. На практике наиболее устойчивыми из них оказались методы Давидона-Флетчера-Пауэлла и Бройдена.

При правильном выборе модели МНК дает высокую точность оценки, но имеет ряд практических недостатков. Он требует знания модельной функции для подгонки (что часто недостижимо), использует для расчетов весь массив данных, на практике ограниченный причинно (4) и исторически (5), не имеет (в общем

случае) предсказуемого времени расчета, не всегда дает однозначный результат (если есть несколько локальных минимумов), не является надежными (нет гарантии сходимости). Поэтому в общем случае параметрические методы и МНК полезны для offline анализа, но менее пригодны для online задач управления.

Методы непараметрического сглаживания пытаются подавить шум, используя лишь качественную априорную информацию о поведении сигнала (непрерывность, монотонность и т.д.) и не требуют знания конкретной параметрической функции, что на практике является достоинством. Их можно разделить на локальные и нелокальные методы; линейные и нелинейные методы; методы для случайной и регулярной сетки [61,62,63,64].

Локальные («оконные») методы используют для оценки $y_{(x)}^*$ не всю выборку $\{x_i, y_i\}$, а только окрестность («окно») вблизи точки x . Так, в качестве окрестности можно взять выборку из точек, попадающих в локальную Δ -окрестность точки x

$$\{x_i, y_i\}_{(x,\Delta)} = \{x_i, y_i\} : |x - x_i| < \Delta \quad (9)$$

Другой вариант выбора окрестности x дает алгоритм k ближайших соседей. Этот метод делает выборку $\{x_i, y_i\}_{(x,k)}$, содержащую k ближайших к точке x узлов x_i .

Заметим, что различие методов проявляется только при нерегулярной сетке x_i . В

случае $\{x_i, y_i\}_{(x,\Delta)}$ ширина окна постоянна, а число точек выборки зависит от

плотности узлов x_i в окрестности точки x . В случае $\{x_i, y_i\}_{(x,k)}$ наоборот число

точек выборки постоянно, а ширина окна меняется. Используя для расчета

небольшую часть данных, локальные методы в общем случае имеют более низкую

точность, чем нелокальные, однако они хорошо приспособлены для задач online

обработки в реальном времени, т.к. работают существенно быстрее нелокальных и

имеют предсказуемое время расчета. Поскольку данные обычно упорядочены по

времени (7), это позволяет даже при случайной сетке x_i (6) делать выборку

$\{x_i, y_i\}_{(x,\Delta)}$ быстрыми алгоритмами двоичного поиска [65]. Кроме того, причинные

(4) и исторические (5) ограничения на выборку не служат помехой для локальных

методов. Поэтому локальные непараметрические методы [64] являются основным

инструментом сглаживания в системах управления [59]. Например, к ним

относятся цифровые и медианные фильтры, методы локального усреднения.

Нелокальные методы используют для оценки $y_{(x)}^*$ весь массив данных $\{x_i, y_i\}$. Например, это Фурье и wavelet анализ, сглаживание сплайнами [61,62]. Эти методы полезны для offline анализа, но для online обработки менее пригодны из-за относительно длительных расчетов, а также причинных (4) и исторических (5) ограничений на выборку, доступную в момент измерений.

Линейные методы отличаются тем, что оценка $y_{(x)}^*$ линейно зависит от измеренных данных y_i . Они хорошо изучены теоретически, дают однозначный и надежный результат, предсказуемое время расчета, поэтому могут использоваться как для offline анализа, так и для online обработки в реальном времени. К ним относятся, например, цифровые фильтры и локальное усреднение.

Нелинейные методы отличаются тем, что оценка $y_{(x)}^*$ нелинейно зависит от измеренных данных y_i . Они меньше изучены теоретически и носят чаще всего эмпирический характер. Примером локального нелинейного метода является медианный фильтр, используемый для подавления локальных «выбросов».

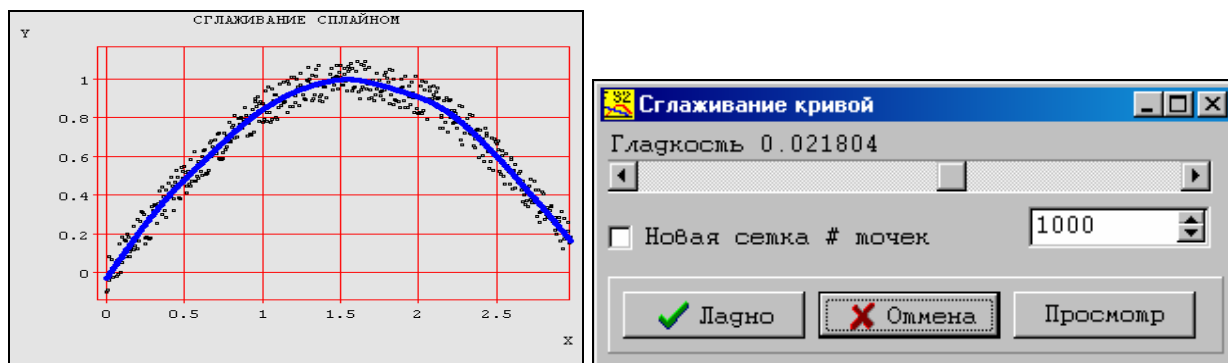


Рис.10. Диалог сглаживающего сплайна.

В пакете CRW-DAQ реализованы сглаживающие сплайны [61,62]. Это однозначный, надежный, линейный, нелокальный метод, имеющий предсказуемое время расчетов, может работать на случайной сетке x_i . Он доступен как набор функций CRW API на языке Object Pascal, а также как готовый инструмент с графическим интерфейсом для диалогового выбора параметра гладкости (Рис.10). Сплайны чаще применяют в offline анализе, а не в задачах АСКУ, поскольку метод нелокальный. Исключением являются калибровки, так как стандартные таблицы калибровок (Таблица 3) в пакете интерполируются сплайнами.

Сглаживающий кубический сплайн ищет оценку $y_{(x)}^*$ в виде функции $f_{(x,\lambda)}$, которая на каждом отрезке $[x_i, x_{i+1}]$, $i = 1..N-1$ представляет полином 3 степени

$$f(x) = S_i(x) = \sum_{k=0}^3 a_{k,i} \cdot (x - x_i)^k \quad (10)$$

и которая дважды непрерывно дифференцируема на интервале $[x_1, x_N]$, то есть принадлежит классу функций $C^2[x_1, x_N]$, и при этом минимизирует функционал:

$$J(f, \lambda) = \lambda \int_{x_1}^{x_N} \left(\frac{d^2 f(x)}{dx^2} \right)^2 dx + \sum_{i=1}^N \frac{1}{\rho_i} (f(x_i) - y_i)^2 \quad (11)$$

где $\rho_i > 0$ - дисперсии в i -й точке. Если они неизвестны, полагается $\rho_i = 1$. Обычно также накладываются граничные условия 2 рода:

$$\frac{d^2 f(x_1)}{dx^2} = \frac{d^2 f(x_N)}{dx^2} = 0 \quad (12)$$

Функционал (11) содержит две части. Первая (интеграл квадрата второй производной) связана с априорной информацией о сигнале и налагает «штраф» за нарушение гладкости. Вторая (взвешенная сумма квадратов невязок) связана с апостериорной информацией и «штрафует» за отклонение от экспериментальных данных. Параметр λ выражает компромисс между априорной и апостериорной информацией о сигнале. При значении $\lambda = 0$ сплайн превращается в интерполяционный, а при $\lambda \rightarrow \infty$ вырождается в константу. Минимизация функционала ведет к 5-диагональной системе линейных уравнений для искоемых коэффициентов сплайна, решение которой ищется методом прогонки [61,62].

Одним из часто встречающихся искажений сигнала являются «выбросы», т.е. импульсные помехи с большой амплитудой. Эффективным средством борьбы с ними является медианный фильтр. Он относится к непараметрическим, локальным, нелинейным и надежным методам, имеет предсказуемое время расчетов, может работать на случайной сетке x_i . В пакете CRW-DAQ реализован пороговый медианный фильтр, доступный в виде функций на языке Object Pascal, а также в виде готового инструмента с графическим интерфейсом (Рис.11).

Алгоритм $2 \cdot M + 1$ - точечного медианного фильтра с абсолютным порогом ε_a и относительным порогом ε_r можно описать так. Для оценки значения $y_{(x_i)}^*$ в точке x_i делается выборка $\{y_k\}_{i,M} = \{y_k : k \in [i - M, i + M], y_{k+1} > y_k\}$, содержащая $2 \cdot M + 1$ точек вокруг x_i и отсортированная по возрастанию значений y . Вычисляется

медиана y_{med} как значение $y_{k=i}$ в середине этой выборки. Если разница $|y_{med} - y_i|$ медианы y_{med} и значения y_i превышает абсолютный или относительный порог, то в качестве оценки берется медиана y_{med} , иначе остается исходное значение y_i :

$$y_{(x_i)}^* = \begin{cases} y_i, & |y_{med} - y_i| \leq \max(\varepsilon_a, |y_i| \cdot \varepsilon_r) \\ y_{med}, & |y_{med} - y_i| > \max(\varepsilon_a, |y_i| \cdot \varepsilon_r) \end{cases} \quad (13)$$

Пороговый медианный фильтр хорошо удаляет «выбросы», не затрагивая неповрежденную часть сигнала, что делает его незаменимым средством обработки данных, содержащих импульсные помехи. Однако его применение в задачах управления ограничено тем, что хотя метод и локальный, он не работает при наличии причинных ограничений (4), требуя наличия данных до и после точки x . В некоторых задачах online обработки медианный фильтр может применяться, если можно смириться с задержкой выдачи результата. Отметим также, что медианный фильтр не позволяет оценить значения вне узловых точек x_i , поэтому он используется в сочетании с одним из интерполяционных методов.

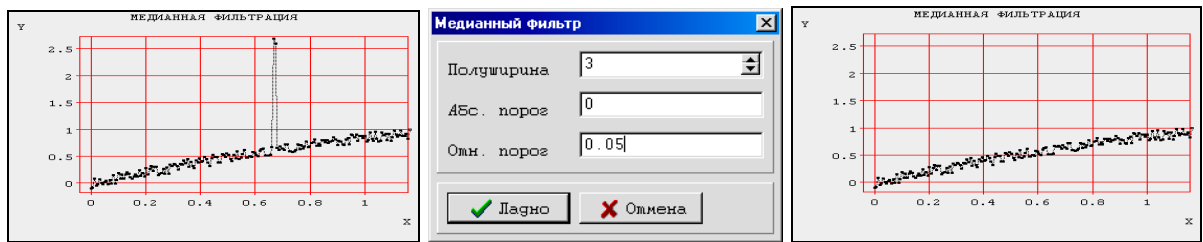


Рис.11. Медианный фильтр и пример его воздействия на сигнал.

На рисунке (Рис.11) показан пример действия медианного фильтра. Сигнал, содержащий импульсную помеху, обрабатывается 7-точечным медианным фильтром с 5% порогом, после чего импульсная помеха исчезает.

Основным инструментом для online сглаживания в пакете CRW-DAQ является набор непараметрических, линейных, локальных методов, работающих на случайной сетке x_i , дающих однозначную, надежную оценку, имеющих предсказуемое время расчета. Это связано со спецификой данных, характерных для распределенных систем управления: причинные (4) и исторические ограничения (5), нерегулярная сетка (6). Хотя набор содержит нескольких методов, он «упакован» в одну функцию с 4 параметрами: $(\Delta, p, k1, k2)$.

Параметр Δ задает полуширину окна сглаживания (9). Поскольку массив данных упорядочен, границы окна сглаживания находятся алгоритмом быстрого двоичного поиска (дихотомии), что резко ускоряет расчеты.

Параметр p задает степень и тип алгоритма. При значении $p = -1$ используется взвешенное среднее интегральное значение. Оно вычисляется как

$$y_{(x)}^* = \frac{\sum_{|x-x_i|<\Delta} 0.5 \cdot (y_{i+1} \cdot w_{(x,x_{i+1})} + y_i \cdot w_{(x,x_i)}) \cdot (x_{i+1} - x_i)}{\sum_{|x-x_i|<\Delta} 0.5 \cdot (w_{(x,x_{i+1})} + w_{(x,x_i)}) \cdot (x_{i+1} - x_i)}, \quad w_{(x,x_i)} = \varphi\left(\frac{x-x_i}{\Delta}\right) \quad (14)$$

где $\varphi(t) \geq 0$, $\varphi(t) = \varphi(-t)$ - некоторая положительная симметричная функция (ядро сглаживания), определенная ниже. Легко увидеть, что формула (14) является конечно-разностной аппроксимацией интегральной формулы свертки с ядром $\varphi(t)$ по методу трапеций:

$$y_{(x)}^* = \frac{\int_{x-\Delta}^{x+\Delta} \varphi\left(\frac{t-x}{\Delta}\right) \cdot y(t) \cdot dt}{\int_{x-\Delta}^{x+\Delta} \varphi\left(\frac{t-x}{\Delta}\right) \cdot dt} \quad (15)$$

При значении $p = 0$ используется взвешенное среднее значение:

$$y_{(x)}^* = \frac{\sum_{|x-x_i|<\Delta} y_i \cdot w_{(x,x_i)}}{\sum_{|x-x_i|<\Delta} w_{(x,x_i)}}, \quad w_{(x,x_i)} = \varphi\left(\frac{x-x_i}{\Delta}\right) \quad (16)$$

На регулярной сетке времени, когда $x_{i+1} - x_i = const$, взвешенное среднее эквивалентно интегральному, различие появляется лишь при неравномерной сетке.

При значении $p > 0$ используется локальная аппроксимация полиномом степени p , рассчитанном по взвешенному МНК. Если точнее, в качестве оценки $y_{(x)}^*$ принимается значение полинома

$$P_{(t,x,c,\Delta,p)} = \sum_{k=0}^p \left(\frac{t-x}{\Delta}\right)^k \cdot c_k \quad (17)$$

в точке $t = x$, с коэффициентами c_k , которые находятся минимизацией взвешенной суммы квадратов по окну сглаживания:

$$c = \arg \min \left(\frac{\sum_{|x-x_i| \leq \Delta} (y_i - P_{(t,x,c,\Delta,p)})^2 \cdot w_{(x,x_i)}}{\sum_{|x-x_i| \leq \Delta} w_{(x,x_i)}} \right), \quad w_{(x,x_i)} = \varphi\left(\frac{x-x_i}{\Delta}\right) \quad (18)$$

При вычислении коэффициентов полинома возникает система линейных уравнений, решаемая модифицированным методом Гаусса. Поэтому время расчета существенно больше, чем при усреднении, хотя и остается предсказуемым. Для гладких сигналов полиномы точнее, чем усреднение, но при наличии в сигнале резких «фронтов» усреднение предпочтительнее, т.к. полиномы могут давать в точках разрыва сигнала осцилляции (эффект Гиббса) [63], опасные для АСКУ, тогда как для усреднения (14,16) характерно «заглаживание» фронтов.

Параметры k_1, k_2 задают функцию ядра:

$$\varphi_{(t)} = \begin{cases} 0, & |t| \geq 1 \\ (1 - t^{k_1})^{k_2}, & |t| < 1 \end{cases} \quad (19)$$

Такой выбор связан с тем, что при разных параметрах k_1, k_2 он дает набор положительных, симметричных, непрерывных, дифференцируемых ядер, имеющих конечное число ненулевых производных. Доказано, что такие ядра имеют оптимальные свойства с точки зрения точности оценки [64]. Положительные ядра выбраны также потому, что они снижают эффект Гиббса [63,64], проявляющийся в виде осцилляций при резком изменении сигнала. Для задач управления такой «дребезг» опаснее, чем «заглаживание» фронта, которое дает положительное ядро.

Пакет CRW-DAQ включает также средства цифровой фильтрации и Фурье-анализа: алгоритмы быстрого (FFT) и дискретного (DFT) Фурье-преобразования, а также цифровые фильтры с конечной импульсной характеристикой (FIR, Finite Impulse Response) [63], доступные в виде функций на языке Object Pascal, а также графических утилит. Эти методы относятся к линейным, быстрым и надежным непараметрическим алгоритмам, причем FFT и DFT являются нелокальным, а FIR – локальным методом. FFT, DFT и FIR фильтры редко использовались в АСКУ, т.к. они требуют сетки x_i с постоянным шагом, что не характерно для распределенных систем управления. Кроме того, они мало пригодны для online обработки, т.к. не могут обрабатывать причинно ограниченные выборки (4). Однако они полезны для offline анализа. Например, Фурье-анализ полезен для анализа спектра шума, что часто позволяет установить его источник и принять меры к его устранению.

1.4.2.3. Компоненты для калибровки измерительных каналов

Практически все современные измерения носят косвенный характер, то есть измеряется не сама интересующая величина, а некая зависящая от нее легко регистрируемая электрическая величина. В пакете CRW-DAQ используется оригинальный подход для проведения и использования калибровок, необходимых для преобразования измеряемых величин в физические.

Калибровка описывает параметрический закон $y = y_{(x,z,c)}$, преобразующий аргумент x , т.е. непосредственно измеряемую величину, в искомую физическую величину y , при данном внешнем параметре z (который может и отсутствовать). Вид закона $y = y_{(x,z,c)}$ определяется типом датчика, а обобщенный параметр c описывает особенности конкретного датчика. Например, при измерении температуры термопарой аргумент x - это термоЭДС [мВ], искомая физическая величина y - это температура термопары [°C], параметр z - это температура холодного спая термопары [°C], зависимость $y_{(x,z,c)}$ - это термопарная кривая, а параметры c - это индивидуальные свойства измерительного канала (коэффициент усиления, смещение нуля АЦП и т.д.).

При калибровочных измерениях получают массив N точек калибровки $\{x_i, y_i, z_i, w_i\}$, $i = 1..N$. Каждая точка калибровки (x_i, y_i, z_i, w_i) описывает i -е калибровочное измерение, в котором непосредственно измеряемая величина x приняла значение x_i , а физическая величина y имела значение y_i , измеренное альтернативным способом. Значение z_i , если оно присутствует, задает внешний параметр, при котором делалось калибровочное измерение. Весовой фактор $w_i > 0$ отражает степень доверия к i -й точке, обычно это обратные величины $w_i = \rho_i^{-1}$ дисперсии ρ_i или единичные веса $w_i = 1$, если дисперсии неизвестны. В качестве альтернативного способа измерения выступают либо физические особые точки, например, точки плавления эталонных образцов, либо показания откалиброванного эталонного датчика, измеряющего ту же величину, что и калибруемый датчик.

Задачу калибровки можно сформулировать как определение оптимального значения параметра c^* в параметрической зависимости $y = y_{(x,z,c)}$, наиболее точно

описывающего калибровочные данные $\{x_i, y_i, z_i, w_i\}$. В качестве критерия точности описания можно принять, например, взвешенную сумму квадратов отклонений (3).

Предположим, что известна пара взаимно - обратных преобразований f_x, f_x^{-1} и f_y, f_y^{-1} (назовем их функциями линеаризации):

$$x' = f_x(x, z), \quad x = f_x^{-1}(x', z) \quad (20)$$

$$y' = f_y(y, z), \quad y = f_y^{-1}(y', z) \quad (21)$$

обладающих тем свойством, что в линеаризованных переменных $\{x', y'\}$ величина y' не зависит от параметра z , а зависимость $y'(x')$ является гладкой функцией и потому может быть удовлетворительно описана полиномом $p_{n,c,x_0,\Delta}(x')$ степени n , с коэффициентами $c = \{c_0, c_1, \dots, c_n\}$, с центром в точке x_0 и масштабом Δ :

$$y' = p_{n,c,x_0,\Delta}(x') = \sum_{k=0}^n c_k \left(\frac{x' - x_0}{\Delta} \right)^k \quad \text{или} \quad y' = p_{n,c}(t) = \sum_{k=0}^n c_k t^k, \quad t = \frac{x' - x_0}{\Delta} \quad (22)$$

Введение параметров x_0, Δ связано с тем, что аппроксимация полиномом хорошо работает на единичном интервале $t \in [0..1]$, поэтому его центр x_0 и масштаб Δ выбираются так, чтобы полином $p_{n,c}(t)$ был определен на этом интервале [65]. Для этого, например, можно положить

$$x_0 = \min\{x'_1, \dots, x'_N\}, \quad \Delta = \max\{x'_1, \dots, x'_N\} - x_0 \quad (23)$$

На практике функции линеаризации определяются типом датчика и измерительного канала. Так, при использовании платинового датчика температуры в качестве функции линеаризации выступает стандартная калибровка Pt-100, известная по опубликованным таблицам ГОСТ. В простейшем случае функции линеаризации могут вообще отсутствовать, то есть задаваться тождественным преобразованием $f_x(x, z) = x, \quad f_y(y, z) = y$.

Введение функций линеаризации необходимо, поскольку не все типы датчиков имеют калибровочные кривые, описываемые полиномом. Так, на кривой калибровки криогенного термодиода (тип D 200 19 707, фирма LEYBOLD AG) ясно виден «излом» при 24 К (Рис.19,а), который нельзя описать полиномом, но после линеаризации зависимость величины от кода АЦП становится линейной.

По физическому смыслу функции линеаризации описывают поведение «идеального» датчика данного типа. Они выбираются по возможности так, чтобы

зависимость приобрела вид тождества $y' = x'$. Реальный датчик отличается от идеального, например, термopара может отклоняться от стандарта из-за вариации её химического состава. Измерительный канал, за счет смещения нуля и усиления, вносит свои отклонения. В первом приближении все эти отклонения приводят к линейной деформации калибровочной кривой $y' = c_0 + c_1 x'$ или полиномиальной $y' = \sum_{k=0}^n c_k (x')^k$. Этим обосновано введение функций линеаризации.

Применив к массиву $\{x_i, y_i, z_i\}$ точек калибровки функции линеаризации, получаем массив линеаризованных калибровочных точек $\{x'_i, y'_i\}$:

$$x'_i = f_x(x_i, z_i), \quad y'_i = f_y(y_i, z_i) \quad (24)$$

Оптимальные коэффициенты c^* калибровочного полинома $p_{n,c,x_0,\Delta}(x')$ подгоняются по МНК с весовыми множителями $w_i > 0$:

$$c^* = \arg \min \sum_{i=1}^N (y'_i - p_{n,c,x_0,\Delta}(x'_i))^2 \cdot w_i \quad (25)$$

или

$$c^* = \arg \min \sum_{i=1}^N \left(f_y(y_i, z_i) - \sum_{k=0}^n c_k \cdot \left(\frac{f_x(x_i, z_i) - x_0}{\Delta} \right)^k \right)^2 \cdot w_i \quad (26)$$

В качестве весов w_i берутся обратные дисперсии измерения $w_i = \rho_i^{-1}$, или полагается $w_i = 1$, если они неизвестны. Задача МНК приводит к системе линейных уравнений, решаемой модифицированным методом Гаусса [65].

После определения коэффициентов c^* полинома искомая калибровочная зависимость приобретает вид

$$y(x) = f_y^{-1}(p_{n,c^*,x_0,\Delta}(f_x(x, z)), z) \quad (27)$$

Применение аппарата калибровок иллюстрирует 2, где приводятся примеры калибровок и соответствующих им калибровочных файлов.

Для описания и хранения калибровок пакет CRW-DAQ использует файлы калибровки **.cal*, в виде текстовых файлов с секционной структурой, подобно **.ini* файлам Windows. В них хранятся лишь массивы измеренных точек калибровки $\{x_i, y_i, z_i, w_i\}$ и параметры аппроксимации (20,21,22), а коэффициенты полинома вычисляются автоматически при загрузке файла. Этот подход хорош тем, что, сделав калибровку один раз, можно легко менять её параметры, не производя

новых дорогостоящих калибровочных измерений и не теряя точность расчетов. Описание файлов калибровки есть в справочной документации пакета.

Функции линеаризации (20,21), описывающие тип датчика при калибровке, в пакете именуют стандартными калибровками (Таблица 3). Они основаны на опубликованных таблицах ГОСТ, интерполируемых сглаживающим сплайном (10). Набор стандартных калибровок легко расширяется, для этого достаточно добавить калибровочные таблицы в конфигурационные файлы пакета, т.к. весь математический аппарат уже готов.

Таблица 3. Стандартные калибровки пакета CRW-DAQ.

№	Имя Шкалы TransformX, TransformY	Комментарий, источник, стандарт.	Функции линеаризации $x' = f_x(x, z), x = f_x^{-1}(x', z)$
1	Line	линейная шкала	$x' = x, x = x'$
2	Log	логарифмическая шкала	$x' = \log(x), x = \exp(x')$
3	SubZ	коррекция смещения нуля	$x' = x - z, x = x' + z$
4	HrAl	термопара хромель-алюмель, тип К ГОСТ Р 50431-92 (МЭК 584-1-77)	$x' = E(x) - E(z), x = E^{-1}(x' + E(z))$ E(x)- термопар.таблица, z-темп.хол.спая
5	HrCp	термопара хромель-копель ГОСТ Р 50431-92 (МЭК 584-1-77)	-- --
6	HrCn	термопара хромель-константан, тип Е ГОСТ Р 50431-92 (МЭК 584-1-77)	-- --
7	RhPt13	термопара платина-родий(13%), тип R ГОСТ Р 50431-92 (МЭК 584-1-77)	-- --
8	RhPt10	термопара платина-родий(10%), тип S ГОСТ Р 50431-92 (МЭК 584-1-77)	-- --
9	RhPt30	термопара платина-родий(30%), тип В ГОСТ Р 50431-92 (МЭК 584-1-77)	-- --
10	WReA1	термопара вольфрам-рений, тип А1 ГОСТ Р 50431-92 (МЭК 584-1-77)	-- --
11	WReA2	термопара вольфрам-рений, тип А2 ГОСТ Р 50431-92 (МЭК 584-1-77)	-- --
12	WReA3	термопара вольфрам-рений, тип А3 ГОСТ Р 50431-92 (МЭК 584-1-77)	-- --
13	NiSi	термопара никель-кремний, тип N ГОСТ Р 50431-92 (МЭК 584-1-77)	-- --
14	CuCn	термопара медь-константан, тип Т ГОСТ Р 50431-92 (МЭК 584-1-77)	-- --
15	FeCn	термопара железо-константан, тип J ГОСТ Р 50431-92 (МЭК 584-1-77)	-- --
16	Pt100	температурный датчик сопротивления Pt 100 Ом, DIN43760	$x' = R(x), x = R^{-1}(x')$ R(x)-таблица Pt-100
17	Ni100	температурный датчик сопротивления Ni 100 Ом, www.minco.com/sensorcalc	$x' = R(x), x = R^{-1}(x')$ R(x)-таблица Ni-100
18	SiDiodeD	кремниевый термодиод, тип D данные фирмы-изготовителя	$x' = D(x), x = D^{-1}(x')$ D(x)-таблица фирмы-изготовителя

Для удобства работы пользователя разработан редактор калибровок (Рис.12).

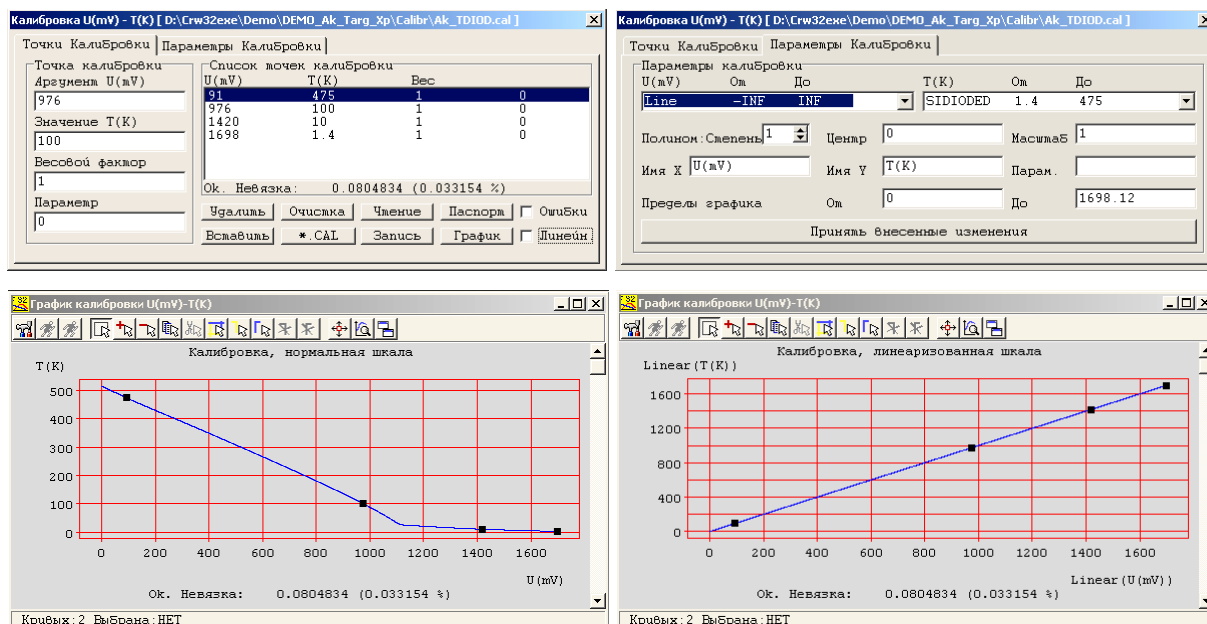


Рис.12. Редактор калибровок пакета CRW-DAQ.

Он позволяет создавать и редактировать калибровки, не вникая в формат файлов калибровки и не делая каких-либо расчетов. В редакторе можно вводить массив точек калибровки, выбирать тип шкалы датчика, параметры аппроксимирующего полинома и сохранять результат в файл. Для контроля качества калибровки используется численная оценка невязки (абсолютная и процентная), а также просмотр графиков кривых калибровки (в исходных и линеаризованных переменных) вместе с точками калибровки. На графиках (Рис.12) видно, что калибровка кремниевого термодиода, сильно нелинейная в исходных переменных, хорошо ложится на прямую линию в линеаризованных переменных.

Развитая система калибровок - важное достоинство пакета, поскольку в других системах решение всех проблем калибровки возлагается на пользователя.

1.4.2.4. Компоненты для интерактивной обработки измеренных данных

Как упоминалось выше, средства интерактивной offline обработки данных привязаны к окнам кривых. Они делятся на две группы: встроенные команды и динамически загружаемые утилиты. Встроенные команды связаны в основном с импортом, экспортом и отображением данных. Часть методов обработки тоже включена во встроенные команды: сглаживающие сплайны (Рис.10), медианный фильтр (Рис.11). Но все же основная часть средств обработки реализована через

динамически загружаемые утилиты двух видов: макросы на языке DAQ Script (3) и динамические расширения DPE (4).

Для связи пакета CRW-DAQ с другими программами (Excel, Word и т.д.) разработаны средства импорта и экспорта данных. Измеренные данные, находящиеся в окнах кривых, можно загружать и сохранять в текстовых таблицах *.txt для обмена с другими программами. Команда печати может не только печатать на принтере, но и передавать данные через буфер обмена в виде таблиц или изображений. Команды табуляции и оцифровки конвертируют данные из окон кривых в текстовые таблицы (и наоборот) для обмена данными с другими программами. Наконец, данными можно обмениваться через DPE расширения, сделав преобразователь форматов на языке Object Pascal.

Интерактивные средства обработки данных подсистемы CRW используются при offline анализе данных. Средства обработки данных для задач управления доступны через программный интерфейс DAQ API, то есть в виде набора функций на встроенных языках программирования, описанных ниже.

1.4.2.5. Программные интерфейсы и языки прикладного программирования

Важными компонентами пакета CRW-DAQ (Рис.5) являются интерфейсы прикладного программирования API (Application Programming Interface). Они позволяют отделить базовое ПО от прикладного и делятся на интерфейсы CRW API (раздел 1.4.2.4) для offline обработки измеренных данных и DAQ API для online обработки, сбора данных и управления АСКУ в реальном времени.

DAQ API содержит программные библиотеки и встроенные языки для прикладного программирования АСКУ – компилятор DAQ Pascal, интерпретатор С-подобного языка DAQ Script, интерпретатор конфигурационных файлов DAQ Config. Программные библиотеки содержат открытый исходный текст функций и процедур на одном из встроенных языков программирования, их можно использовать в прикладных программах. Языки прикладного программирования – это трансляторы (интерпретаторы и компиляторы) исходного кода прикладных программ в непосредственно исполняемые машинные инструкции.

Пакет CRW-DAQ включает нескольких языков программирования, сильно различающихся по возможностям, решающих разные задачи и потому взаимно дополняющих друг друга. Благодаря наличию этих языков, средств редактирования

и отладки пакет содержит полноценную среду разработки и исполнения для создания АСКУ, не требующую для своей работы установки дополнительного ПО.

Интерпретатор конфигураций DAQ Config

Для описания параметров АСКУ служат конфигурационные файлы с расширением CFG, а для чтения этих файлов разработан интерпретатор DAQ Config. В отличие от обычных интерпретаторов он описывает не исполняемые инструкции, а структуры данных и параметры загружаемой АСКУ: теги и кривые, прикладные программы и драйверы, окна и мнемосхемы.

Файлы конфигураций - это текстовые файлы с секционной структурой, подобной INI файлам Windows. В каждой АСКУ есть главный загружаемый конфигурационный файл, который при необходимости ссылается на другие файлы конфигураций, программ, мнемосхем, калибровок и т.д.

Несмотря на сходство с INI файлами, возможности CFG файлов значительно выше, т.к. в них секции могут разбиваться на части, расположенные в любом месте файла или в нескольких файлах, а при загрузке одноименные секции объединяются в одну. Специальная секция *[ConfigFileList]*, содержащая список «*ConfigFileList = ...*», позволяет включать одни конфигурации в состав других.

Каждая строка секции конфигурации может содержать декларацию, переменную, ссылку или список. Декларация вводит в состав конфигурации новый объект, создаваемый в момент ее загрузки, например, декларация

```
[DeviceList]  
&DimSrv = device software program
```

создает новое устройство (прикладную программу) с именем *&DimSrv*. Обычно за декларацией следует детальное описание созданного объекта в одноименной с ним секции, в данном случае *[&DimSrv]*.

Переменные задают значения параметров АСКУ, включая параметры декларированных объектов. Их описание имеет простой формат «*Имя = Значение*» или «*Имя = Значение1, Значение2,...*». Ссылки отличаются от переменных только тем, что они содержат в качестве значений имена файлов конфигураций, программ, тегов и т.д. для последующей ссылки на них. Например, описание

```
[&DimSrv]  
InquiryPeriod = 10  
ProgramSource = ..\DaqPas\DimSrv.pas
```

задает значение периода опроса устройства *InquiryPeriod* в миллисекундах, а также ссылку на файл *ProgramSource* с исходным кодом программы на языке DAQ Pascal. Ссылки обычно содержат относительные имена файлов, что дает возможность переноса каталога конфигурации АСКУ в любое место. Базой, относительно которой задаются имена файлов, обычно служит главный конфигурационный файл.

Списки имеют формат «Имя = Элемент1, Элемент2,...» и могут, в отличие от переменных, быть многострочными, например, описание

```
[Windows]
Plot.PT = Curve_Window
[Plot.PT]
CurveList = T1, T2, T3
CurveList = P1, P2, P3, P4
```

декларирует окно кривых *Plot.PT* и создает список *CurveList* из семи кривых для отображения в этом окне.

Полное описание файлов конфигураций, мнемосхем и калибровок есть в справочной документации пакета [19], доступной после его инсталляции.

Интерпретатор команд и макросов DAQ Script

Пакет CRW-DAQ включает встроенный C-подобный интерпретатор DAQ Script, разработанный на языке Object Pascal. Интерпретатор мал размером и хорошо защищен от ошибок в прикладном программном коде. За счет интерпретации он может выполнять инструкции, неизвестные в момент компиляции пакета, например, выражения, вводимые пользователем в режиме диалога. Например, DAQ Script используется при вычислении формул в интерактивном режиме, когда требуется их немедленная интерпретация.

В данном пакете работает множество экземпляров интерпретатора, каждый из которых можно рассматривать как виртуальную машину со своим пространством переменных, констант, функций и команд. В DAQ Script легко добавлять новые функции и команды, поэтому все интерпретаторы имеют единый синтаксис, но сильно различаются функциональностью за счет разного набора встроенных функций и команд. Описание разработанного интерпретатора приведено в приложении (см. Приложение 3).

Компилятор и виртуальная машина DAQ Pascal

Основным средством разработки прикладных программ АСКУ в пакете CRW-DAQ является язык DAQ Pascal, созданный на основе классического

свободно распространяемого компилятора Pascal-S Николауса Вирта [67]. Он написан на языке Object Pascal. В основе его реализации лежит несколько идей: виртуальная машина (VM), JIT (Just In Time) компиляция «на лету», ясность и безопасность языка. Рассмотрим структуру DAQ Pascal (Рис.13).

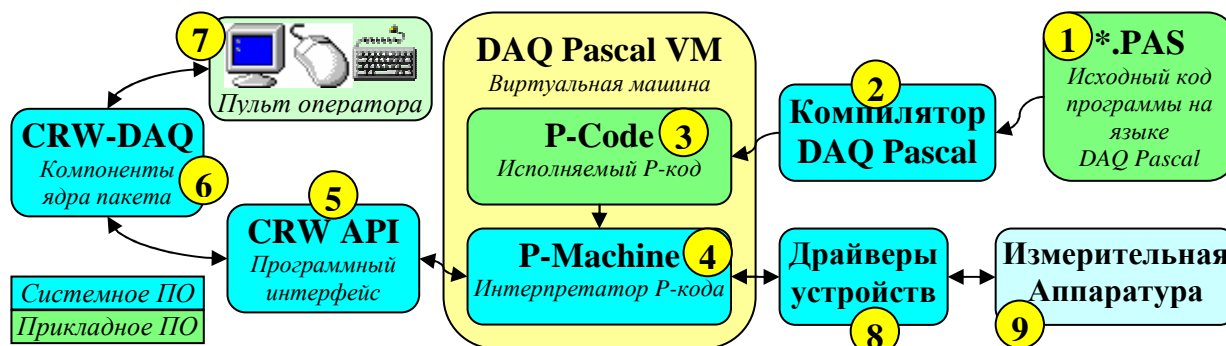


Рис.13. Структура компонентов DAQ Pascal.

Исходный код прикладной программы на языке DAQ Pascal (1) в момент загрузки АСКУ транслируется JIT-компилятором DAQ Pascal (2) в промежуточный код (P-Code) для некоторой платформенно-независимой виртуальной машины (3), который затем выполняется на данной платформе интерпретатором P-кода (4). При необходимости интерпретатор P-кода с помощью программного интерфейса CRW API (5) обращается к функциям ядра пакета (6), реализующим платформенно-зависимые функции, такие как графический интерфейс для связи с оператором (7) или драйверы (8) для управления измерительной аппаратурой (9).

Идея JIT-компиляции состоит в том, что компилятор не создает промежуточных исполняемых файлов (EXE или DLL), а генерирует исполняемый P-код виртуальной машины прямо в памяти. Недостатком является замедление загрузки АСКУ, хотя транслятор работает довольно быстро, компилируя $\sim 10^4$ строк в секунду. Достоинством является простота (не надо поддерживать форматы исполняемых файлов и хранить их), надежность системы и платформенная независимость прикладных программ, исполняемых на виртуальной машине.

Идея виртуальной машины VM (Virtual Machine) состоит в том (Рис.14), что исполняемым кодом является не исходный код на языке высокого уровня (ЯВУ), и не платформенно-зависимый машинный код, а «промежуточный» код абстрактной виртуальной машины, исполняемый на данной платформе VM-интерпретатором (3). С точки зрения скорости VM – промежуточное решение между «чистым» интерпретатором (1), декодирующим и сразу исполняющим команды ЯВУ и

«чистым» компилятором, создающим «натуральный» исполняемый машинный код (2). Для типичного процессора с тактовой частотой $\sim 1\text{GHz}$ скорость интерпретатора будет $\sim 10^5$, VM-интерпретатора $\sim 10^7$, а «натурального» кода $\sim 10^8$ операций в секунду (по порядку величины).

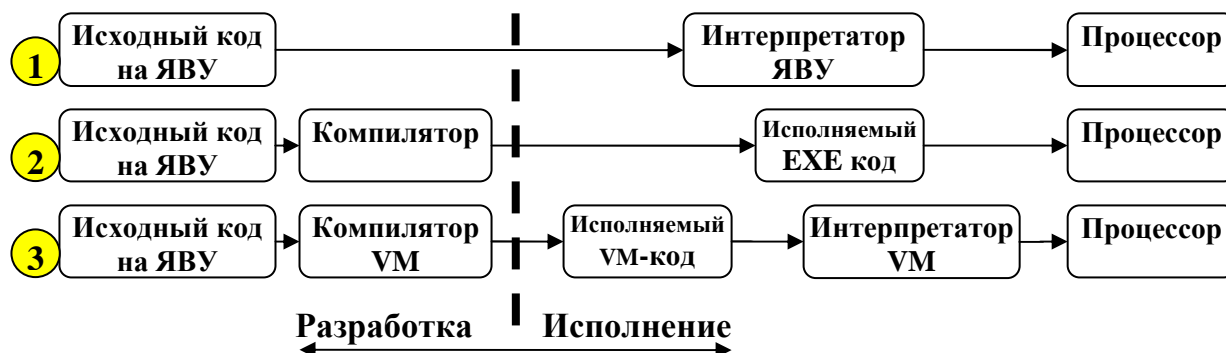


Рис.14. Идея интерпретатора (1), компилятора (2) и виртуальной машины (3).

Снижение скорости работы программ при использовании VM оправдано повышением надежности и отказоустойчивости прикладных программ при неизбежных сбоях и ошибках. VM-интерпретатор (Рис.14), стоящий между исполняемым кодом и процессором, дает возможность изолировать прикладные программы друг от друга для локализации возможных ошибок, а также эффективно защитить прикладной код при программных ошибках или сбоях в измерительной аппаратуре, приводящих к появлению недопустимых входных данных. Так, при возникновении типичной ошибки выхода индекса массива за допустимый диапазон вместо аварийного завершения всей программы произойдет лишь сбой работы данной прикладной программы, а остальные - продолжат нормальную работу. В сочетании с многопоточностью (каждая VM выполняется в отдельном потоке) это приводит к повышенной «живучести» АСКУ: сбой в одной подсистеме не ведет к сбою системы в целом и потере её управляемости. VM-интерпретатор может также контролировать использование прикладной программой системных ресурсов (памяти, файлов, сетевых каналов), предотвращая их «утечку». Это повышает долговременную стабильность работы АСКУ. Реализованная защита VM-интерпретатора оказалась весьма эффективной: аварийных остановок работы пакета по вине ошибок в прикладных программах DAQ Pascal не наблюдалось, хотя ошибки в прикладном коде, конечно, неоднократно возникали.

Язык разработки прикладных программ управления АСКУ, создаваемых специалистами самой разной квалификации, должен быть простым, однозначно

интерпретируемым и не должен содержать «лишних» и «опасных» конструкций. Поэтому его реализация была ограничена рамками классического процедурного языка Pascal [60]. Главным средством расширения языка выбраны встроенные функции, так как они не усложняют компилятор и VM-интерпретатор, и не снижают степень защиты ПО.

Созданный автором большой набор встроенных функций (~400) для доступа к ядру пакета, данным RTDB, измерительной аппаратуре и другим функциям CRW API делает DAQ Pascal мощным инструментом разработки АСКУ. Полный список встроенных функций и их описание есть в справочной системе пакета. Функции делятся на группы: арифметические, тригонометрические и специальные функции; кодирование, декодирование и обработка строк; файловые операции; измерение времени; обработка текстов; конфигурирование, отладка и обмен сообщениями; работа с устройствами DAQ системы; обработка кривых и тегов; работа с мнемосхемами и окнами; обмен данными по каналам связи RS-232, RS-485 и сокетам TCP/IP; работа с портами ввода-вывода ISA, PCI; работа с процессами; работа с динамическими расширениями DPE в виде загружаемых библиотек DLL.

Богатый набор функций времени позволяет организовывать периодические события по расписанию и задержки, измерять длинные и короткие интервалы времени, контролировать время работы алгоритмов управления.

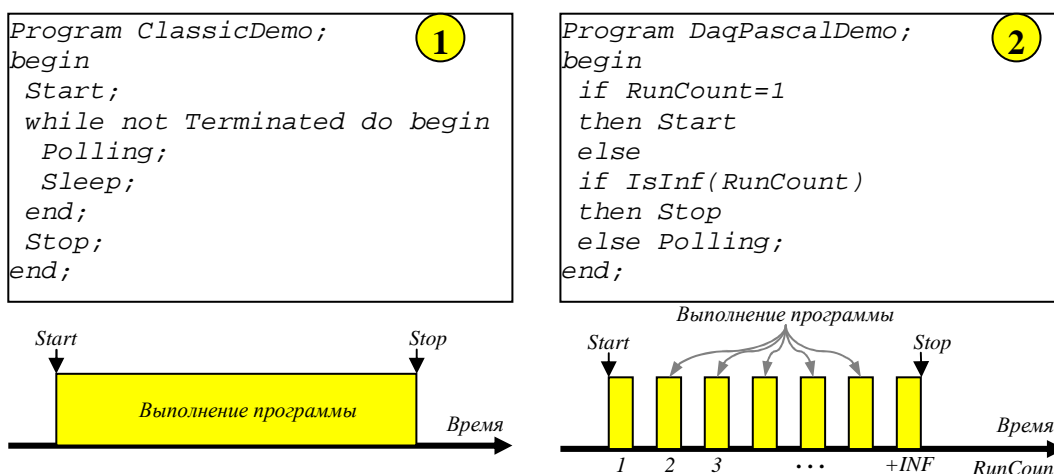


Рис.15. Различие логики обычных программ (1) и программ DAQ Pascal (2).

Важным отличием DAQ Pascal от других языков является сама логика выполнения прикладных программ (Рис.15). Обычные программы (1) рассчитаны на однократное исполнение. После запуска и инициализации (*Start*) они входят в бесконечный цикл обработки очереди событий (*Polling*), освобождая процессор в

конце каждого цикла (*Sleep*). Цикл прерывается командой завершения программы (*Terminate*), она освобождает ресурсы (*Stop*) и завершается. При следующем вызове программа «не помнит» результат предыдущего сеанса и начинает работу «с нуля».

Напротив, программы DAQ Pascal (2) рассчитаны на многократное выполнение с периодическим вызовом по системному таймеру, причем переменные программы сохраняют свои значения между вызовами. Это делает программы DAQ Pascal похожими на процедуры обработки прерываний системного таймера, хотя реализованы они иначе. При первом после старта АСКУ вызове программы функция счетчика вызовов *RunCount* принимает значение 1, позволяя выполнить инициализацию (*Start*). При остановке АСКУ вызов *RunCount* возвращает *+INF*, позволяя по условию *IsInf(RunCount)* выполнить завершающие действия (*Stop*). При остальных вызовах выполняется обработка событий (*Polling*).

Программы DAQ Pascal по своей логике должны не содержать замкнутых циклов, а быстро выполнять требуемые действия по управлению устройствами и сразу завершаться, освобождая процессор до следующего такта таймера. Поэтому вместо задержек или замкнутых циклов ожидания событий программы DAQ Pascal обычно используют работу по «флагам» и переменным состояниям, то есть пишутся как набор «правил», состоящих в выполнении определенных действий при возникновении некоторых условий. Например, наступление ожидаемого события часто сигнализируется изменением значения какого-то тега по сравнению с предыдущим вызовом. Для иллюстрации стиля прикладных программ DAQ Pascal возьмем простейший пример – организацию периодического (раз в секунду) вызова процедуры опроса *Polling*. На языке DAQ Pascal это выглядит примерно так:

```
program PeriodicDemo;
const Period=1000; var LastTime:Real;
begin
  if msecnow-LastTime >= Period then begin
    LastTime:=msecnow;
    Polling;
  end;
end.
```

При каждом вызове программы функция времени *msecnow* (миллисекунд от Р.Х.) сравнивается с оставшимся от прошлых вызовов временем *LastTime* предыдущего выполнения *Polling*, и если прошло более *Period* миллисекунд, время запоминается в *LastTime* и выполняется *Polling*.

Описанный способ работы DAQ Pascal программ, который из-за отказа от использования задержек можно назвать «асинхронным» стилем программирования, хорошо подходит для решения задач управления АСКУ в реальном времени. Алгоритм управления разделяется на кванты с четкой привязкой по времени. В программах отсутствуют задержки, а поэтому снижается риск их «повисания» из-за неопределенно долгого ожидания событий. Сохранение значений переменных между вызовами позволяет легко отслеживать динамику управляемых процессов. Появляется возможность контролировать частоту опроса потоков и определять «повисшие» потоки. Все это способствует повышению надежности АСКУ.

Если для решения задачи функциональности или скорости DAQ Pascal не хватает, есть возможность вызова созданных на языке Object Pascal динамических расширений DPE (раздел 1.4.2.4) в виде DLL файлов, дающих полный доступ ко всем возможностям компьютера (см. Приложение 11, Листинг 2).

В DAQ Pascal реализована библиотека встроенных функций для работы с процессами, обеспечивающих создание, запуск и завершение процесса, а также возможность двустороннего обмена данными с созданным дочерним процессом по анонимному каналу связи через функции стандартного консольного ввода-вывода, переназначенные на ввод-вывод в канал (см. Приложение 11, Листинг 3).

Выделение отдельных задач в консольные программы, взаимодействующие с пакетом CRW-DAQ через программу DAQ Pascal и анонимный канал, является одним из хороших способов добавления новых драйверов и служб. Достоинством такого подхода является облегчение создания прикладных программ (нет ничего проще консольных приложений) и повышение надежности АСКУ (добавляемая программа работает в отдельном процессе и не снижает надежность пакета).

Для работы с портами ввода-вывода, аппаратными интерфейсами и сетевыми протоколами ISA, PCI, RS-232, RS-485, CAN, Ethernet, TCP/IP реализована большая библиотека встроенных функций, позволяющая создавать на языке DAQ Pascal драйверы для локально или удаленно подключаемых устройств, в дополнение к набору встроенных в ядро пакета драйверов (см. Приложение 11, Листинг 4).

Уникальным свойством DAQ Pascal является возможность создания и отладки прикладных программ и драйверов в реальном времени, в процессе измерений. Объединенная среда разработки и исполнения пакета CRW-DAQ и

многопоточный режим работы позволяют редактировать и компилировать программы DAQ Pascal в процессе измерений. Поток прикладной программы приостанавливается, не затрагивая другие потоки, компилятор создает новый P-код программы, затем поток возобновляет выполнение уже с новой программой. Это ускоряет разработку драйверов устройств и управляющих программ АСКУ.

Если сравнивать DAQ Pascal с оригинальным компилятором Pascal Н.Вирта, то мало изменился лишь синтаксический анализатор и генератор P-кода. Интерпретатор P-кода переписан автором практически полностью, а библиотека встроенных функций создана автором целиком. Кроме того, автор интегрировал компилятор и VM-интерпретатор в систему сбора данных и реализовал «асинхронную» логику выполнения программ DAQ Pascal.

Интегрированный компилятор DCC языка Object Pascal

В дополнение к встроенным интерпретаторам в CRW-DAQ интегрирован компилятор Object Pascal, генерирующий быстрый «натуральный» машинный код.

Встроенный компилятор DCC имеет две сферы применения - динамические расширения DPE в виде DLL библиотек и консольные драйверы в виде EXE файлов. Динамические расширения применяются при offline обработке данных (см. раздел 1.4.2.4) и в задачах сбора данных и управления (см. раздел 1.4.2.6).

Консольные драйверы – это приложения, запускаемые из программы DAQ Pascal как дочерний процесс (обычно в скрытом окне) и взаимодействующие с ней через анонимный канал связи, переназначенный вместо консоли ввода-вывода. Они оказались хорошим средством реализации аппаратных драйверов устройств и служб, таких как генератор речевых сообщений или Web сервер. Их достоинством является повышение надежности АСКУ, так как сбой в консольном драйвере, выполняемом в отдельном процессе, не приводит к общему сбою системы. Кроме того, консольные драйверы легко отлаживать, используя на этапе отладки простой консольный ввод-вывод, не переназначая его в канал.

1.4.2.6. Компоненты для разработки и выполнения прикладного ПО АСКУ

Для разработки ПО АСКУ и его поддержки на этапе выполнения в пакете CRW-DAQ создан ряд компонентов (Рис.5), объединенных названием «DAQ-система». Компонент DAQ (Data Acquisition) содержит программные

средства, непосредственно связанные со сбором данных и управлением, и является основным инструментом при создании ПО АСКУ. Компонент RTDB (Real Time Data Base) содержит базу данных реального времени, в которую входят теги и кривые. Теги – поименованные скалярные переменные, отражающие текущее значение некоторой величины. Кривые – поименованные динамические массивы данных $\{t_i, y_i, i=1..N\}$, которые обычно представляют исторические данные, то есть зависимости $y(t)$ измеряемых величин y от времени t . Компонент DAQ Control содержит средства организации и управления системой сбора данных, включая интерфейс пользователя для запуска и остановки измерений, конфигурирования и диагностики DAQ системы. Компонент DAQ Runtime содержит исполнительную систему сбора данных и управления, включая устройства, драйверы, виртуальные машины прикладных программ на встроенном языке DAQ Pascal или DAQ Script, средства калибровки каналов и сжатия измеренных данных.

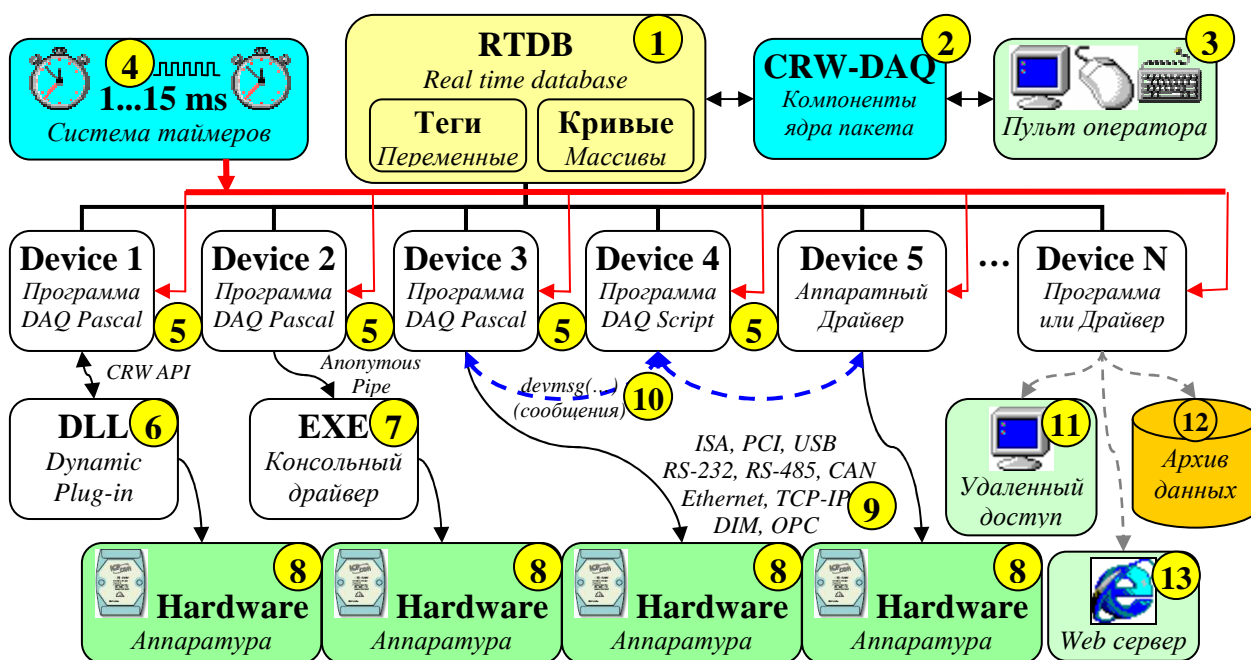


Рис.16. Структурная схема DAQ-системы.

Центральными понятиями DAQ-системы (Рис.16) являются: хранилище данных реального времени RTDB и устройства (Device). Устройства входят в состав компонента DAQ Runtime, который поддерживает работу DAQ-системы на этапе выполнения. Компонент DAQ Control осуществляет общее управление DAQ-системой (загрузка, запуск, остановка, завершение и т.д.).

RTDB (1) является централизованной базой данных - поименованных тегов (скалярных переменных) и кривых (динамических массивов) для хранения

измеряемых величин. Она оптимизирована по времени доступа к данным (~100 ns) и хорошо защищена (в смысле потоковой безопасности). Компоненты ядра пакета CRW-DAQ (2) используют хранилище RTDB для отображения данных и диалога с оператором (3). Собственно сбор данных и управление осуществляют устройства (5), тактируемые системой таймеров (4).

Под устройством (Device) в DAQ-системе понимается некоторый исполняемый код (прикладная программа или драйвер) вместе с необходимыми для его работы данными, который выполняет свою четко определенную задачу, например, управляет физической аппаратурой (8), подключенной через аппаратный интерфейс (9). Устройство может содержать встроенный в пакет аппаратный драйвер для управления измерительными модулями, или прикладную программу на встроенных языках DAQ Script, DAQ Pascal. Прикладная программа DAQ Pascal при необходимости может также загружать динамическое расширение в виде DLL модуля (6) или запускать в дочернем процессе консольный драйвер (7), обмениваясь с ним данными по анонимному каналу связи (pipe).

Каждое устройство (кроме драйверов измерительных модулей в сети RS-485) выполняется в своем программном потоке и работает автономно, напрямую не связано с другими устройствами, но имеет доступ к хранилищу данных RTDB. Устройства изолированы друг от друга и могут взаимодействовать только двумя способами: изменением совместно используемых тегов RTDB или посылкой друг другу сообщений *devmsg* (10). Довольно необычна реализация этих сообщений. Все они - строковые, помещаются в консоль стандартного ввода прикладной программы и считываются процедурой *readln*, т.е. обработка сообщений состоит в анализе консольного ввода. Это сильно облегчает разработку и отладку программ, т.к. все сообщения можно эмулировать через связанное с устройством консольное окно, также помещающее данные в консоль стандартного ввода.

Алгоритм работы DAQ-системы можно упрощенно описать следующим образом. Имеется хранилище данных (тегов и кривых RTDB) и набор прикладных программ (устройств), работающих параллельно и независимо друг от друга, связанных с RTDB и с измерительной аппаратурой. По тактовым сигналам от системы таймеров прикладные программы в каждом кванте времени считывают

данные из хранилища или аппаратуры, анализируют и обрабатывают их и затем записывают обработанные данные в хранилище или исполнительные устройства.

Тактирование происходит по квантам времени планировщика потоков Windows с периодом от 1 до 15 мс, причем в этом случае система приоритетов программных потоков позволяет высокоприоритетным потокам сбора данных вытеснять низкоприоритетные служебные потоки. Важно, что алгоритмы управления отделены от системы таймеров, что позволяет легко менять принцип тактирования и контролировать частоту опроса устройств.

Обсуждение вопросов обеспечения безопасности, надежности и отказоустойчивости измерительных систем и вопросы их конфигурирования в пакете CRW-DAQ вынесены в приложение (см. Приложение 1,5,6,7,8,9,10).

1.4.2.7. Средства для создания распределенных систем управления

Пакет CRW-DAQ поддерживает создание распределенных систем управления. Они могут быть необходимы: при большом геометрическом размере установки, потребности в удаленном управлении при радиационной опасности, для повышения надежности системы за счет иерархии управления (Рис.62), для снижения шумов и наводок путем размещения измерительных модулей вблизи источника сигнала и передачи измеренных данных по цифровой линии связи.

Распределенные системы управления строятся двумя путями. Первый путь состоит в создании многомашинных систем из компьютеров в сети Ethernet. Для межмашинной связи в пакете используется основанная на протоколе TCP/IP технология DIM (Distributed Information Manager) [70], разработанная в ЦЕРН для проведения экспериментов на ускорителях. Она реализует высокоуровневый сетевой механизм взаимодействия между процессами, ориентированный на распределенные системы управления реального времени. В состав пакета входит и WEB сервер для обмена данными по протоколу HTTP, позволяя использовать для удаленного наблюдения и управления обычный WEB-браузер. Приложение 7 рассматривает вопрос использования технологий DIM и WEB более подробно.

Вторым путем организации распределенных систем управления являются сети интеллектуальных контроллеров и модулей удаленного сбора данных на базе интерфейсов RS-232, RS-485, CAN. В пакет включены драйверы измерительных

модулей и модулей управления серии I-7000, I-87000 (ICP-DAS), ADAM-4000 (Advantech), выбранных по следующим причинам:

- выпускается широкая номенклатура многоканальных модулей аналогового и дискретного ввода/вывода с открытым протоколом обмена данными, имеющих сертификаты РФ, в том числе для использования в атомной промышленности;
- на базе модулей возможно построение легко масштабируемых распределенных систем на основе помехоустойчивой сети RS-485 (скорость до 115200 бит/сек) в радиусе до 1 км (без репитера) от управляющего компьютера;
- модули имеют высоковольтную гальваническую изоляцию (~3 кВ) и обеспечивают эффективную фильтрацию электромагнитных помех, позволяют снизить уровень наводок за счет коротких сигнальных кабелей при их монтаже вблизи источника сигнала и передаче данных по сети RS-485;
- модули аналогового ввода имеют высокую точность измерений (~0,1%, 16 бит), хорошую стабильность и скорость оцифровки (от 1 до 100 Гц/канал), а модули дискретного ввода/вывода обеспечивают достаточно быструю реакцию (порядка 10 мс) при дискретном контроле и управлении;
- модули созданы специально для работы в тяжелых условиях промышленности и транспорта и обеспечивают высокую надежность и отказоустойчивость, допускают «горячую» замену, обладают аппаратным сторожевым таймером (*watchdog*), автоматически перезапускающим модуль в случае его «зависания», и программным сторожевым таймером, переводящим все выходы модуля в заранее определенные «безопасные» состояния в случае «зависания» управляющего компьютера или обрыва связи с ним;
- модули отличает высокая технологичность при работе: надежный монтаж под винт, минимум разъемных соединений, прочный пластиковый корпус, защищенный от пыли и влаги, монтаж на стене или на DIN-рейке;
- модули относительно дешевы в связи с большим тиражом их выпуска.

В пакете CRW-DAQ автором реализована система прав доступа и защиты, необходимая для предотвращения аварийных ситуаций, могущих возникнуть из-за несанкционированных действий посторонних лиц при локальном или удаленном доступе к системе управления (см. Приложение 9).

1.5. Автоматизация комплекса подготовки газовой смеси и системы радиационного контроля

Комплекс подготовки газовой смеси КПГС [6] обеспечивает подготовку смеси ИВ заданного состава и чистоты, её подачу в мишень, вакуумирование газовых коммуникаций, утилизацию тритийсодержащей газовой смеси, очистку системы от следов трития и его соединений. Источником ИВ служат металлогидридные генераторы [76], очистку от примесей выполняют диффузионные палладиевые фильтры [77]. При их работе надо измерять и стабилизировать температуру нагрева, отключать нагрев при превышении заданных температур и давлений, контролировать обрывы в цепях нагрузки, блокировать включение нагревателей при определенных состояниях вентилей или при опасном повышении уровня радиации. Для управления нагревом создана многоканальная программно-аппаратная подсистема регуляторов, аналогичная описанной в разделе 3.3.2, которая обеспечивает высокую надежность управления и точность стабилизации температуры не хуже $\pm 2^\circ\text{C}$.

Подсистема контроля и управления КПГС (Рис.17) включает 7 таких каналов регулирования для генераторов BS1, BS2, BS3, BS4 и фильтров F1, F2, F3. Управление нагревом ведется через активную мнемосхему, позволяющую задавать температуру нагрева и условия блокировки по температурам и давлениям (Рис.51).

Для контроля вакуума в узлах КПГС используются четыре манометрических термопарных преобразователя ПМТ-4М и два датчика вакуума IKR261, TPR260 фирмы Balzers. Лампы ПМТ подключены к электронному блоку, регулиющему и контролирующему их ток. Выходное напряжение термопарных преобразователей ламп измеряется многоканальным модулем АЦП I-7018, давление вакуума вычисляется по калибровке (35). Датчики вакуума Balzers подключаются через 6-канальный контроллер TPG-256, управление контроллером и сбор цифровых данных по вакууму осуществляются через порт RS-232.

Высокие давления измеряются тензометрическими датчиками, а низкие (до 10÷20 Бар) - датчиками типа САПФИР. Тензометрические датчики питаются от стабилизатора с напряжением +12 В, выходные сигналы датчиков регистрируются модулями АЦП I-7018. Для преобразования зарегистрированных напряжений в давления используются полиномиальные калибровки (раздел 1.4.2.3).

Состояния вентилей (VP) и электромагнитных клапанов (VE) установки (110 датчиков положений, по 2 на каждый ventиль и клапан) контролирует адаптер дискретного ввода-вывода DIO-144 на шине ISA управляющего ПК. Сигналы датчиков поступают через коммутационный блок, преобразующий их в TTL. Состояния четырех датчиков давления воды контролируются многоканальным модулем дискретного ввода I-7053. Программа АСКУ обеспечивает отображение состояний вентилей, клапанов и датчиков давления воды на экране компьютера.

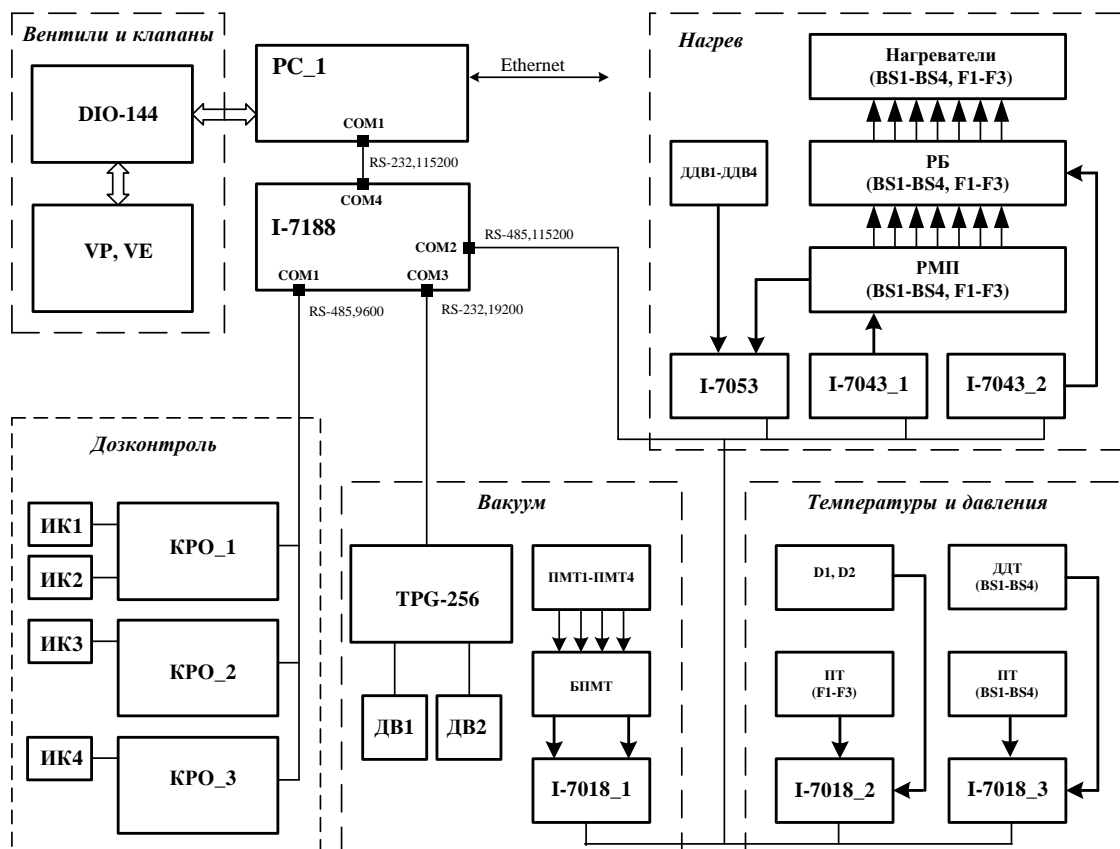


Рис.17. Структурная схема АСКУ подсистемы КПГС и СРК.

PC_1- управляющий компьютер подсистемы; I-7188- PC-совместимый микропроцессорный контроллер; DIO-144- 144-канальный ISA-адаптер дискретного ввода/вывода; VP, VE- контролируемые ventили и электромагнитные клапаны; КРО- микропроцессорные контроллеры радиационной обстановки; ИК- ионизационные камеры; TPG-256- 6-канальный контроллер для вакуумных датчиков фирмы Balzers; ДВ- датчики вакуума фирмы Balzers; D- датчики давления САПФИР-22-Ех-М; ПМТ- лампы типа ПМТ-4; БПМТ- блок подключения ламп ПМТ; ПТ- преобразователи термопарные; РМП- регуляторы мощности полупроводниковые; РБ- реле блокирующие; ДДТ- датчики давления тензометрические; I-7018- 8-канальный модуль аналогового ввода; I-7043- 16-канальный модуль дискретного вывода; I-7053- 16-канальный модуль дискретного ввода; ДДВ- датчики давления воды.

АСКУ КПГС и СРК базируется на PC_1 (Рис.17). Модули серии I-7000, контроллеры датчиков вакуума TPG-256 и контроллеры радиационной обстановки имеют разные интерфейсы и скорости передачи данных, поэтому для их подключения к управляющему ПК используется интеллектуальный контроллер I-7188, снабженный управляющей программой на языке Turbo Pascal 7.0.

Контроллер 7188 на базе PC-совместимого микропроцессора имеет 4 порта последовательного ввода-вывода и выполняет функцию адресуемого преобразователя интерфейса. Порт COM4 (RS-232) используется им для связи с управляющим компьютером. Безадресный контроллер TPG-256 подключаются к порту COM1 модуля I-7188 и, с точки зрения управляющего ПК, имеет собственный виртуальный адрес. Модули серии I-7000 подключаются к порту COM2 (RS-485, 115200 бит/сек), а двухканальные измерительные блоки контроля радиационной обстановки - через порт COM3 (RS-485, 9600 бит/сек).

При проектировании комплекса газового обеспечения особое внимание уделялось вопросам радиационной безопасности при работе с тритием. Безопасность обеспечивается как путем использования принципов физической защиты при построении газовой системы установки, так и подсистемой дозиметрического контроля по тритию.

Для измерения объемной активности трития в газовых коммуникациях и воздухе рабочей зоны служат автоматизированные многоканальные радиометры трития РГБ-06МА с промышленными блоками детектирования БДГБ-02П. РГБ-06МА (сертификат RU.C.38.046.A №11019 от 01.11.2001) обеспечивает измерение ОА в диапазоне от $5 \cdot 10^5$ до $5 \cdot 10^{10}$ Бк с ИК объемом 1 л, от $5 \cdot 10^4$ до $5 \cdot 10^9$ Бк с ИК объемом 10 л. Предел погрешности измерения ОА трития не более $\pm 25\%$.

Радиометр работает как автономно, так и в составе автоматизированной многоканальной системы радиационного мониторинга [58], и обеспечивает измерение объемной активности трития с учетом её фонового уровня, двухпороговый анализ состояния радиационной обстановки по каждому из каналов и управление двумя стандартными внешними блоками сигнализации БСР-19П.

Программа управления КПГС обеспечивает контроль аналоговых (вакуум, температуры, давления) и дискретных (состояния вентилей, клапанов, датчиков давления воды) параметров комплекса. Контролируемые аналоговые параметры отображаются в текстовой и графической форме в режиме реального времени. Графические окна для отображения аналоговых данных обладают широкими возможностями для просмотра и обработки динамических кривых (выбор кривых для просмотра, масштабирование, фильтрация и т.д.). Программа предусматривает непрерывную запись контролируемых параметров на жесткий диск компьютера.

Отображение состояний узлов комплекса и управление установкой осуществляется через активные графические мнемосхемы (Рис.8), которые имеют вид схемы газовых коммуникаций установки и потому интуитивно понятны операторам.

ПО подсистемы радиационного мониторинга обеспечивает управление измерительными блоками КРО, включая установку режима его работы, задание верхнего и нижнего порогов для индикации состояния радиационной обстановки и других параметров. Оно также осуществляет непрерывный контроль состояния радиационной обстановки и сохраняет регистрируемые параметры на жестком диске компьютера, визуально отображает радиационную обстановку на графической панели, подает звуковое предупреждение персоналу при возникновении нештатных ситуаций – например, при превышении заданных порогов объемной активности.

1.6. Управление сменными мишенями и измерение параметров газовой смеси

При изучении мюонного катализа в Н/Д/Т смесях надо проводить измерения в широком диапазоне температур (от 20 до 1600 К) и давлений (до 160 МПа). Для этого используют сменные мишени – жидко-третиевую мишень [53], третиевую мишень высокого давления [54], дейтериевую мишень высокого давления [55]. Подсистема АСКУ мишени работает со всеми типами мишеней [8]. В ее функции входит: контроль вакуума; измерение высоких температур в двух точках ТМВД и ДМВД термодатчиками; измерение высокого давления тензометрическим датчиком; измерение низкого давления датчиком САПФИР; измерение низких температур в двух точках ЖТМ термодиодными датчиками; управление нагревателем с мощностью до 1 кВт и нагревателем малой мощности для работы при низких температурах; стабилизация температуры в рабочем объеме мишени.

АСКУ мишени базируется на компьютере РС_2 (Рис.18). Вакуум контролирует датчик фирмы Balzers через двухканальный контроллер TPG-252. Высокое давление измеряется тензометрическим датчиком, со стабилизированным питанием +12 В, а низкое - датчиком САПФИР. Выходные сигналы датчиков регистрируют модули АЦП I-7018. Для преобразования измеренных напряжений в давления используются полиномиальные калибровки датчиков (раздел 1.4.2.3).

Для нагревателя большой мощности Н1 используется канал управления, описанный в разделе 1.5. Он включает термопары ПТ1 или ПТ2, регулятор мощности РМП_1 и реле блокировки Р1. ТермоЭДС термопар измеряет модуль АЦП I-7018_1. Управление регулятором РМП_1, релейной блокировкой и контроль тока выполняет модуль дискретного ввода/вывода I-7050.

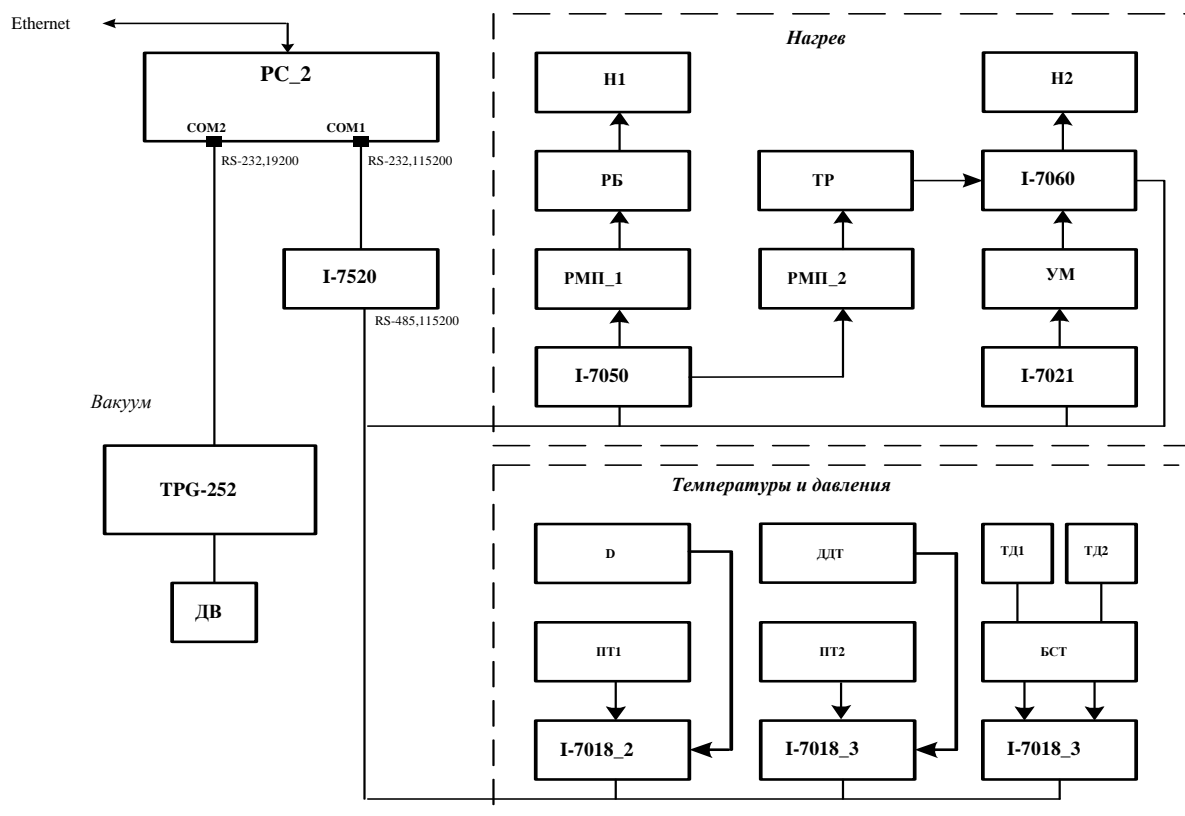


Рис.18. Блок-схема подсистемы контроля и управления мишени.

Н1- нагреватель большой мощности, до 1 кВт; РМП- регулятор мощности полупроводниковый; РБ- реле блокировки нагревателя Н1; Н2- нагреватель малой мощности, до 300 Вт; УМ- усилитель мощности для нагревателя Н2; ТР- трансформатор; ПТ- преобразователь термопарный; ТД- термодиод; ДДТ- датчик давления тензометрический; Д- датчик давления САПФИР; ДВ- датчики вакуума фирмы Balzers; БСТ- блок стабилизаторов тока для термодиодов; I-7520- модуль преобразователя интерфейса RS-232/RS-485; I-7018- 8-канальный модуль аналогового ввода; I-7060- многоканальный модуль релейного вывода; I-7021- 16-разрядный модуль аналогового вывода; I-7050- многоканальный модуль дискретного ввода-вывода; TPG-252- 2-канальный контроллер для вакуумных датчиков фирмы Balzers.

Температура в рабочем объеме мишени регулируется нагревателем Н2 с высокой точностью. Схема позволяет подключать нагреватель Н2 к выходу усилителя мощности (УМ) постоянного тока (мощность до 40 Вт, напряжение до 20 В, ток до 2 А), либо к выходу регулятора РМП_2 через понижающий трансформатор. Регулировка нагрева с помощью УМ нужна при криогенных температурах и выполняется 16-разрядным модулем ЦАП I-7021.

Два кремниевых термодиодных датчика измеряют температуры в диапазоне от 2 К до 330 К. Точность измерения ± 1 К в диапазоне 2-100 К и 1% выше 100 К.

Термодиоды питают стабилизаторы тока $10 \text{ мкА} \pm 0,05\%$, выходные напряжения термодиодов измеряют модули АЦП I-7018. При вычислении температуры используется калибровка с фирменной паспортной температурной характеристикой термодиода (Рис.19,а) в качестве линеаризующей функции (раздел 1.4.2.3), поэтому удается добиться высокой точности калибровки, несмотря на её сильную нелинейность. Указанная аппаратура и ПО подсистемы управления мишенью позволяют задавать и стабилизировать температуру с точностью $\pm 0,05 \text{ К}$.

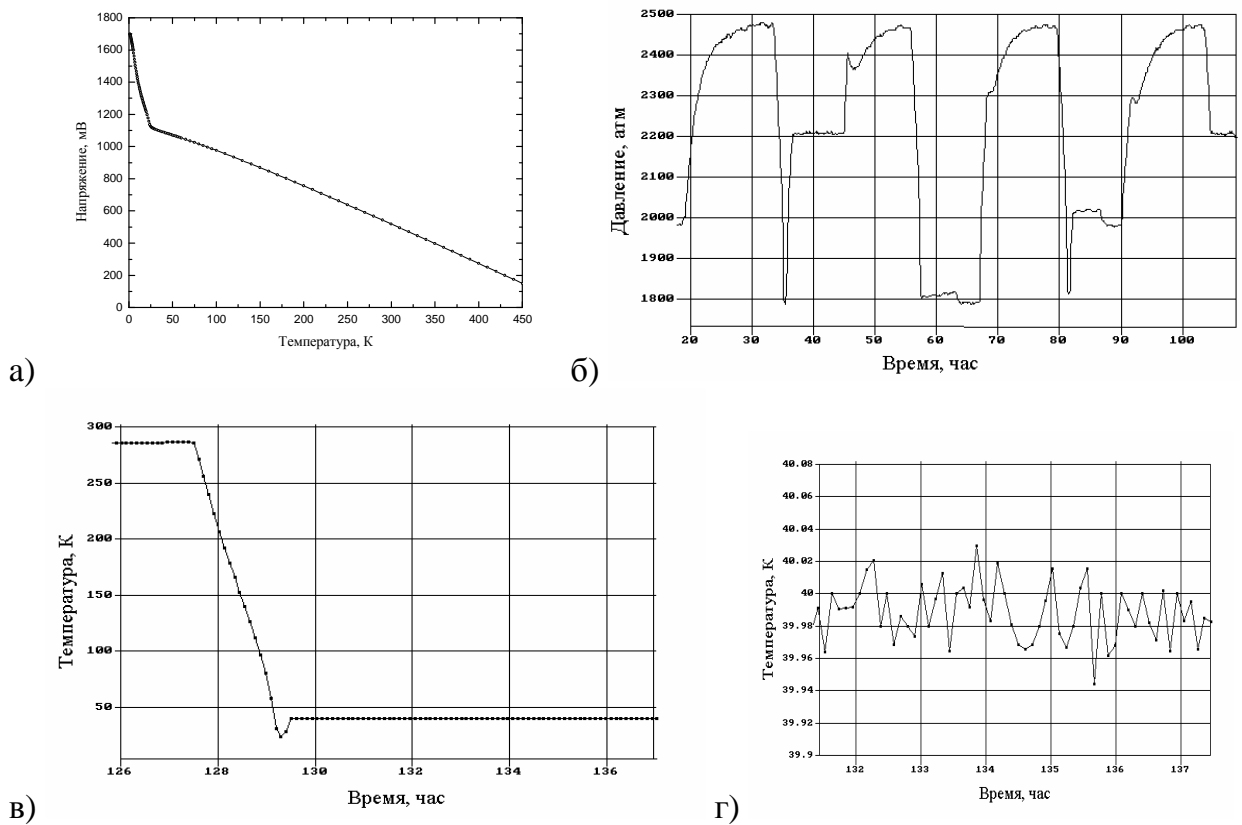


Рис.19. Графики температурной характеристики термодиода (а), кривой давления в тритиевой мишени высокого давления (б), кривой температуры охлаждения криогенной мишени (в), той же кривой на участке стабилизации температуры мишени $40 \pm 0,05 \text{ К}$.

Для получения большей мощности нагреватель H_2 можно переключить на регулятор мощности РМП_2, работающей от сети переменного тока 220 В. Управление регулятором, релейной блокировкой и контроль тока обеспечивает модуль дискретного ввода-вывода I-7050, переключение выполняется программно модулем релейного вывода I-7060. Выбирая коэффициент понижения напряжения трансформатора, можно в широких пределах менять максимальную мощность H_2 .

ПО подсистемы мишени позволяет задавать и стабилизировать температуру нагрева мишени, измерять вакуум, температуры и давления в газовой системе. Для

примера приведены температурные кривые охлаждения и стабилизации для криогенной мишени ЖТМ (Рис.19,в,г) и кривые давления (Рис.18,б) для ТМВД.

1.7. Анализ молекулярного состава газовой смеси

Молекулярный состав газовой смеси измеряется с помощью детектора теплопроводности (катарометра) и малогабаритной (5 см³) ионизационной камеры (ИК). Изотопы разделяются в хроматографической колонке длиной 3 м при температуре жидкого азота (Рис.20). В качестве газа носителя взят неон [56,57].

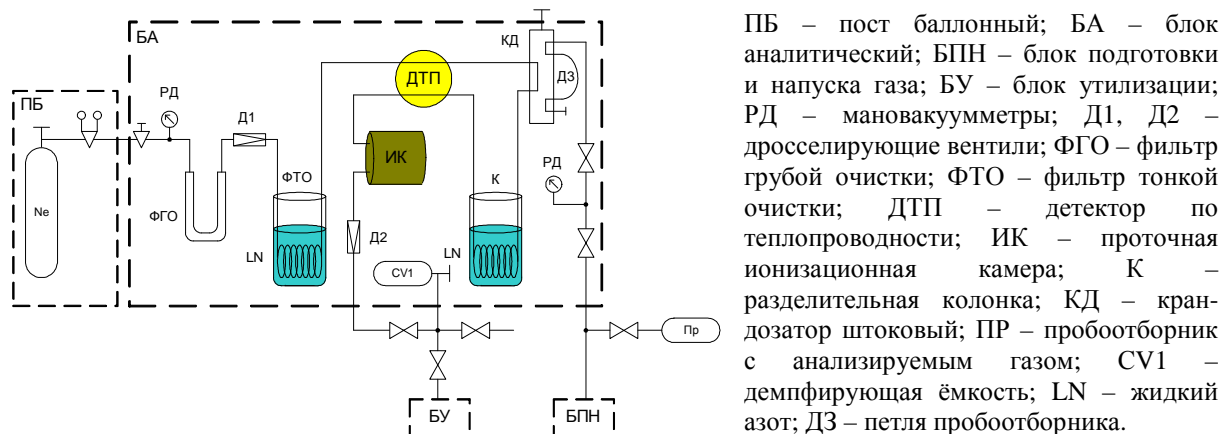


Рис.20. Принципиальная схема газового радиохроматографа для анализа состава Н/Д/Т смеси.

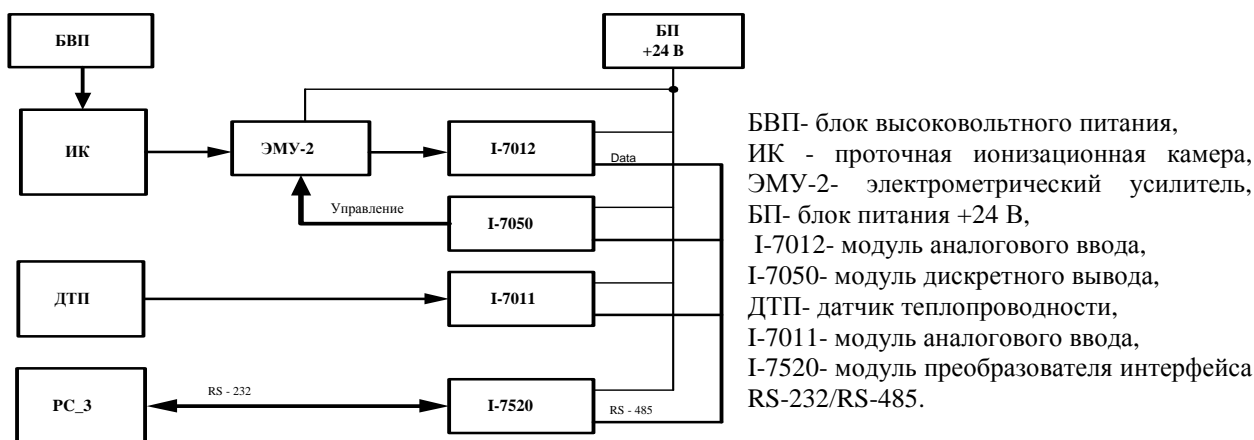


Рис.21. Блок-схема АСКУ подсистемы контроля молекулярного состава смеси.

АСКУ для контроля состава смеси базируется на компьютере PC_3 (Рис.21). Выходной сигнал катарометра регистрирует модуль АЦП I-7011 с частотой ~10 Гц в диапазоне ±50 мВ и точностью ~0,05%. Ток ИК (от 10⁻¹³ до 10⁻⁵ А) измеряется с помощью электрометрического усилителя (ЭМУ) с переключением диапазонов. Выходной сигнал ЭМУ регистрирует модуль АЦП I-7012. Диапазоны ЭМУ переключает модуль дискретного вывода I-7050. Модули подключены к управляющему компьютеру с помощью преобразователя интерфейса RS-232 в RS-485 типа I-7520. Обмен данными выполняется со скоростью 115200 бод.

Шум на самом чувствительном диапазоне ЭМУ составляет около $2 \cdot 10^{-14}$ А и падает до $3 \cdot 10^{-15}$ А при сглаживании фильтром с шириной окна в 1 сек (Рис.22,4).

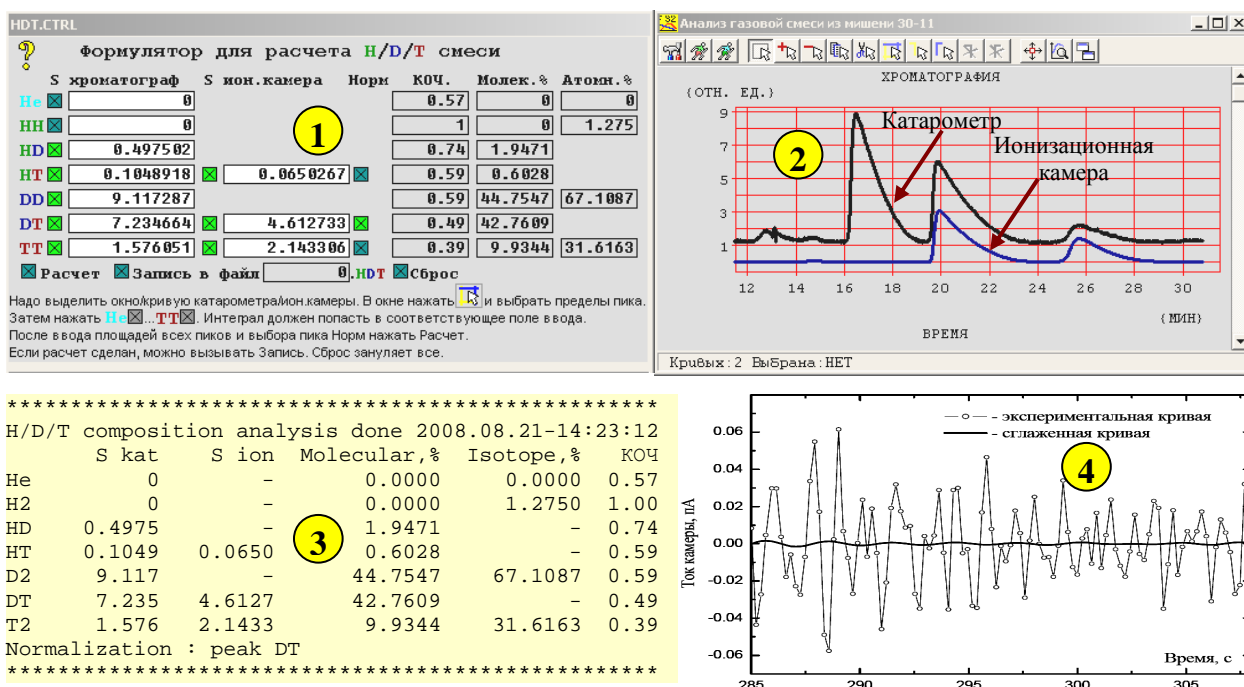


Рис.22. Внешний вид программы анализа состава Н/Д/Т смеси (1), графиков катарометра и ионизационной камеры (2), пример файла с результатом анализа состава смеси (3) и шумовая характеристика измерительного канала электрометрического усилителя (4).

Для анализа молекулярного состава газовой смеси разработана программа (Рис.22,1), обеспечивающая управление сбором данных, сглаживание и обработку аппаратурных кривых катарометра и ионизационной камеры (Рис.22,2), а также расчет молекулярного и изотопного состава Н/Д/Т смеси, который производится следующим образом [56,57]. Дано 6 пиков катарометра {*HH*, *HD*, *HT*, *DD*, *DT*, *TT*}. На первом шаге доля C_i для i -го компонента смеси вычисляется по нормировке:

$$C_i = \frac{K_i \cdot S_i^{kat}}{\sum_{k=HH}^{TT} K_k \cdot S_k^{kat}} \cdot 100 \% \quad (28)$$

где S_i^{kat} - площадь пика катарометра для i -го компонента; K_i - коэффициенты относительной чувствительности (КОЧ) катарометра для i -й молекулы, которые определяются по относительным площадям пиков катарометра в калибровочном опыте при прохождении одинаковых количеств газовых проб из разных молекул. Они оказались равны $K_{HH}/K_{HD}/K_{HT}/K_{DD}/K_{DT}/K_{TT}=1.00/0.74/0.59/0.59/0.49/0.39$.

Чувствительность ИК к содержанию трития выше, чем у катарометра, что позволяет повысить точность измерения состава. Поэтому на второй итерации идет

перенормировка, т.е. коррекция долей $\{HT, DT, TT\}$ по пикам ИК. Она делается по пику n с максимальной амплитудой (и точностью расчета площади) по формуле:

$$C_i^{cor} = \begin{cases} C_n^{kat} \cdot \frac{C_i^{ion}}{C_n^{ion}}, i = HT, DT, TT \\ C_i^{cor} = C_i^{kat}, i = HH, HD, DD \end{cases}, C_i^{ion} = \frac{S_i^{ion}}{K_i^{ion}}, i = HT, DT, TT \quad (29)$$

где C_i^{ion} - концентрация i -го пика по кривой ионизационной камеры; K_i^{ion} - КОЧ ионизационной камеры, равный $K_{HT}/K_{DT}/K_{TT}=0.5/0.5/1.00$; n – пик, выбранный для коррекции доли $\{HT, DT, TT\}$. Из (29) видно что, пики $\{HH, HD, DD\}$ не корректируются. После коррекции повторяется нормировка, результатом будет:

$$C_i^{res} = \frac{C_i^{cor}}{\sum_{k=HH}^{TT} C_i^{cor}} \cdot 100 \% \quad (30)$$

Программа анализа состава газа написана на языке DAQ Pascal (раздел 1.4.2.5) и интегрирована в АСКУ, что позволяет делать online анализ в процессе измерений. Площади пиков катарометра и ИК вычисляются автоматически, оператор лишь идентифицирует пик и задает маркером пределы ROI для расчета площади пика, а затем заносит её в поле ввода нажатием кнопки на мнемосхеме формулятора. Результат анализа сохраняется в виде таблицы в текстовом файле (Рис.22,3).

1.8. Результаты экспериментов на установке ТРИТОН

В 1997-2004 гг. на автоматизированной установке ТРИТОН на фазотроне ЛЯП ОИЯИ с участием автора было проведено более 80 экспериментов по исследованию процесса мюонного катализа в D/T смесях [25]. Изучение МК шло в широком диапазоне параметров смеси – от 0.2 до 1.2 LHD по плотности (в единицах плотности жидкого водорода $LHD=4,25 \cdot 10^{22} \text{ат/см}^3$), от 20 до 800 К по температуре, от 15% до 86% по содержанию трития. Каждой точке диаграммы (Рис.23) соответствует 4÷5 измерений при различных концентрациях ИВ в исследуемой смеси и экспозициях 6 ÷ 12 часов (Рис.24). Все операции по подготовке газовых смесей, анализу их состава, заправке мишени и стабилизации её температуры, утилизации отработавших тритий-содержащих смесей и радиометрическому контролю воздуха проводились под управлением АСКУ комплекса, работавшей непрерывно и безотказно на всех этапах эксперимента. Анализ полученных на установке ТРИТОН данных позволил определить основные

кинетические параметры МК. Они находятся в хорошем согласии с результатами других исследовательских групп (PSI, RIKEN-RAL, LAMPF) в тех областях, где параметры смеси были похожими. Сравнение результатов измерений с принятой теорией МК показывает её эффективность, хотя полное и точное описание процесса МК еще не достигнуто.

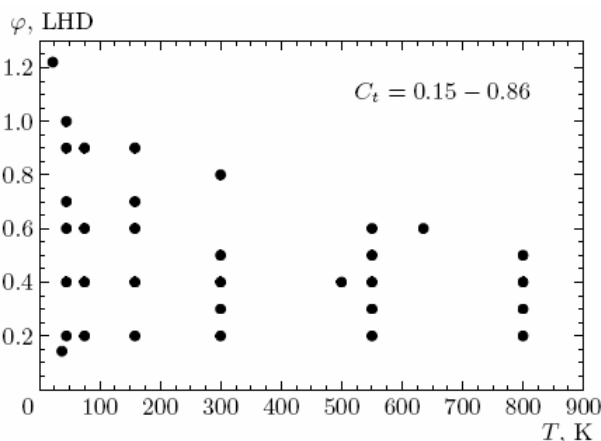


Рис.23. Область параметров φ , T , c_t для МК экспериментов в смесях ИВ, проведённых в 1997-2004 гг. [25].

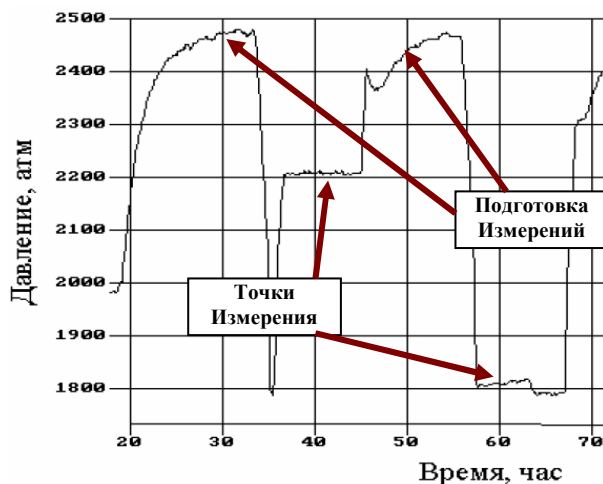


Рис.24. Пример графиков давлений мишени, с обозначением точек измерений.

По результатам измерений [25] впервые была получена зависимость числа циклов МК от концентрации трития в газовой D/T -смеси при температурах от 37 до 800 К и плотностях от 0,143 до 1,024 LHD (Рис.25). Столь широкий диапазон условий систематически исследован впервые. Полученные данные (Рис.26) позволяют оптимизировать условия протекания реакции синтеза при МК, что необходимо для будущих применений МК в энергетике и научных исследованиях.

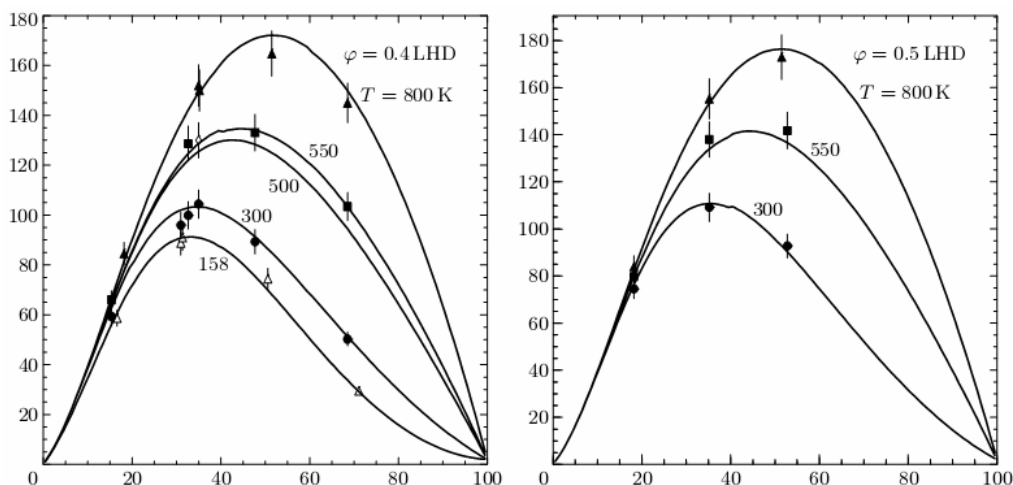


Рис.25. Пример полученных результатов измерений [25]: среднее число циклов МК на один мюон при различных параметрах (плотность, температура, содержание трития) D/T смеси.

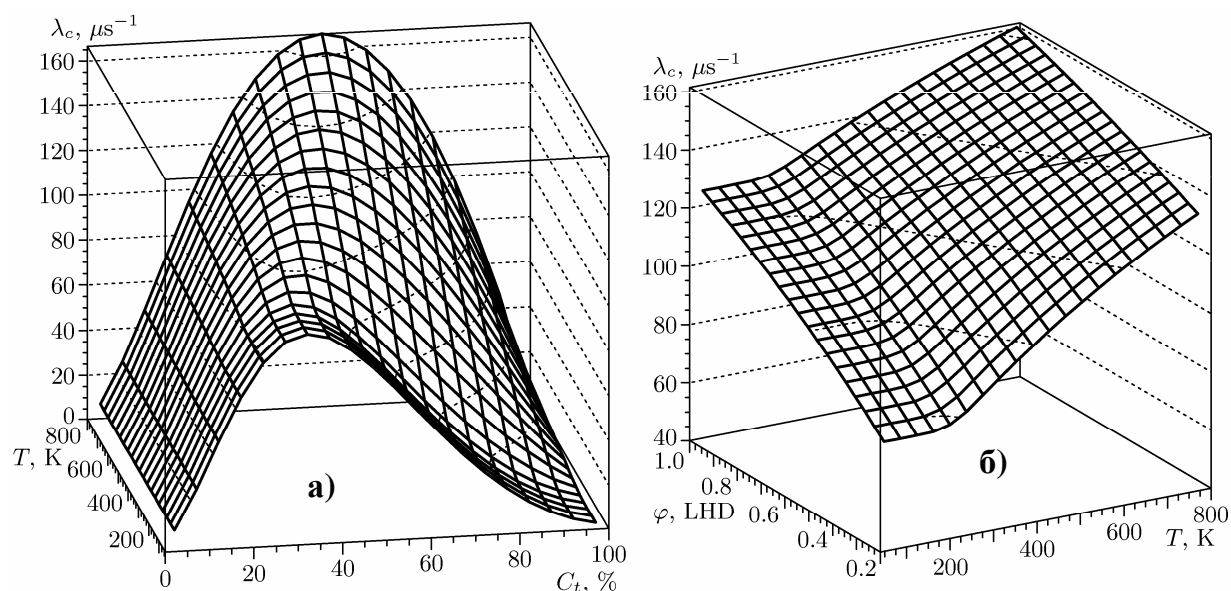


Рис.26. Пример результатов измерений на установке ТРИТОН [25]: аппроксимация зависимости среднего числа циклов МК в D/T смеси а) как функция температуры и концентрации трития при $\varphi=0.4$ LHD, б) как функция температуры и плотности при концентрации $C_t=0.35$.

Для газообразного дейтерия на установке ТРИТОН были проведены экспериментальные исследования температурной зависимости скорости $\lambda_{dd\mu}$ образования $dd\mu$ -молекулы при температурах $300 \div 800$ К и давлениях до 150 МПа [28]. Полученные данные при температурах до ~ 400 К (Рис.27) хорошо совпадают с данными LAMPF. Данные при температурах выше 400 К были получены впервые.

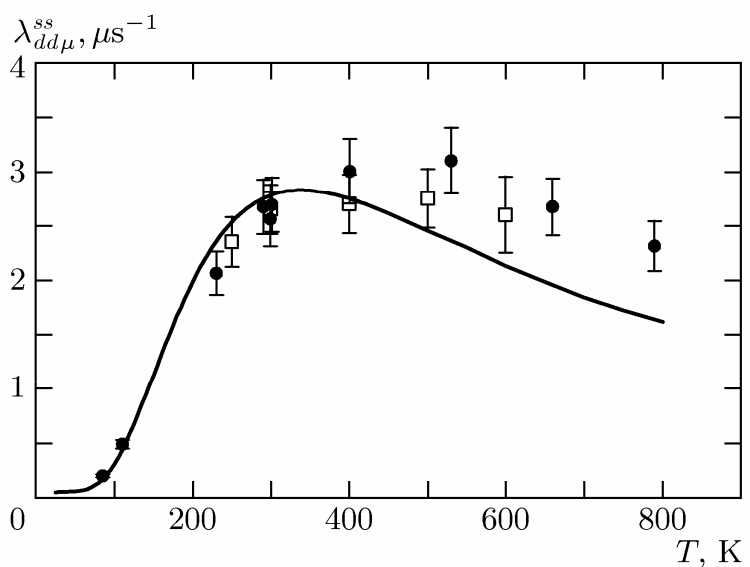


Рис.27 Зависимость $\lambda_{dd\mu}^{F=1/2}$ от температуры [28]: • - ТРИТОН, □ - LAMPF, — - расчеты

Теоретическое предсказание [5] высокой скорости образования $dt\mu$ молекулы на HD молекулах стимулировало интерес к исследованию МК в тройных H/D/T смесях. Эксперименты на установке ТРИТОН по изучению МК при высоких

плотностях в жидких и газообразных тройных $H/D/T$ смесях [26,27,30] показали, что количество актов dt -синтеза на один мюон уменьшается при добавлении в D/T смесь протия. При этом высокая скорость образования $dt\mu$ -молекул, предсказанная теоретиками и подтвержденная прямыми экспериментами, не компенсирует потерь мюона за счёт высокой вероятности его прилипания к гелию в $H/D/T$ смеси.

МК дает уникальные возможности для изучения ядерных реакций, в том числе для изучения реакции радиационного захвата $d+d \rightarrow {}^4\text{He}+\gamma$, которая участвует как в первичном, так и звездном нуклеосинтезе. Малое сечение реакции затрудняет ее исследование в пучковых экспериментах. В экспериментах на дейтериевой мишени высокого давления на установке ТРИТОН была исследована реакция радиационного захвата дейтрона в $dd\mu$ молекуле [32]. Впервые получена экспериментальная оценка выхода этой реакции из состояния $J=1$ $dd\mu$ молекулы на уровне $\eta_\gamma \leq 2 \cdot 10^{-5}$ на один акт синтеза.

Следует заметить, что полученные на установке ТРИТОН результаты исследований процессов МК имеют важное научное и практическое значение, например, при создании мюонно-каталитического гибридного реактора и интенсивного источника монохроматичных нейтронов с энергией 14 МэВ [2].

1.9. Выводы

Для автоматизации тритиевых установок разработан инструментальный программный пакет CRW-DAQ [19,47,48,50,51], позволяющий эффективно создавать ПО для автоматизации исследовательских установок. Он обеспечивает полный цикл разработки, отладки и поддержки АСКУ, позволяет в короткие сроки создавать высоконадежные многопоточные, многопроцессные и многомашинные распределенные системы управления. Его главными достоинствами являются простота освоения, высокая скорость разработки прикладных программ для автоматизации измерений, их высокая безопасность и отказоустойчивость.

С использованием пакета создано программное обеспечение многомашинной распределенной автоматизированной системы контроля и управления тритиевым комплексом высокого давления установки ТРИТОН, предназначенной для исследований процессов МК ядерных реакций в смесях ИВ в широком диапазоне температур, плотностей и концентраций ИВ. Разработанная с участием автора

система измерений и управления тритиевого комплекса обеспечила подготовку и измерение макроскопических параметров (температура, давление, молекулярный и изотопный состав Н/D/T смеси), играющих ключевую роль при обработке и анализе физических результатов измерений.

АСКУ комплекса [8] обеспечивает измерение большого числа аналоговых и дискретных параметров и управление узлами комплекса, измерение изотопного и молекулярного состава смеси изотопов водорода, измерение и стабилизацию параметров мишени в процессе измерений, радиометрический контроль объемной активности трития в газовых линиях и рабочих помещениях.

Подобную АСКУ имеет лишь система тритиевого обеспечения экспериментов по мюонному катализу [78], созданная в Резерфордской лаборатории (Великобритания) примерно в это же время. Однако используемые там количество трития (1,5 кКи), давление в мишени (ниже атмосферного) и рабочая температура мишени (12–23 К) существенно ниже, чем на установке ТРИТОН (до 100 кКи трития в связанном состоянии и до 10 кКи в свободном состоянии, давление в мишени до 160 МПа, температура от 20 до 800 К).

Созданный автоматизированный тритиевый комплекс установки ТРИТОН использовался при проведении большого цикла измерений, в которых получены уточненные и новые данные для параметров цикла МК в дейтерии, в двойной D/T смеси и в тройной Н/D/T смеси в широком диапазоне температур, давлений и концентраций [25,26,27,29,33]. Впервые получены кинетические параметры МК в широком диапазоне плотностей, температур и концентраций трития ($0.2 \div 1.2$ LHD, $20 \div 800$ К, $15 \div 86\%$), измерена скорость образования $dd\mu$ -молекулы при температурах более 400 К [28], получена экспериментальная оценка выхода реакции радиационного захвата дейтрона в $dd\mu$ молекуле [32].

Структура автоматизированной системы контроля и управления позволила оперативно адаптировать ее к изменениям при подготовке к экспериментам, когда происходило обновление или ввод новой аппаратуры, датчиков и каналов управления. Созданное программное обеспечение (пакет CRW-DAQ) и накопленный методический опыт могут эффективно использоваться при создании новых исследовательских установок в области тритиевых технологий.

2. Управление системами для подачи изотопов водорода циклотрона У-400М и комплексом тритиевой мишени на установке АКУЛИНА

2.1. Установки АКУЛИНА

Исследование структуры экзотических легких ядер, находящихся на границе нейтронной стабильности, имеет фундаментальное значение в ядерной физике и представляет одно из центральных направлений в изучении ядерной структуры. В 2000 г. сотрудниками ЛЯР ОИЯИ и РФЯЦ-ВНИИЭФ начата подготовка экспериментов по получению и изучению нуклонно-нестабильных изотопов водорода ${}^4\text{H}$ и ${}^5\text{H}$, образующихся в реакциях $t+t \rightarrow {}^5\text{H}+p$, $t+t \rightarrow {}^4\text{H}+d$ и $t+d \rightarrow {}^4\text{H}+p$.

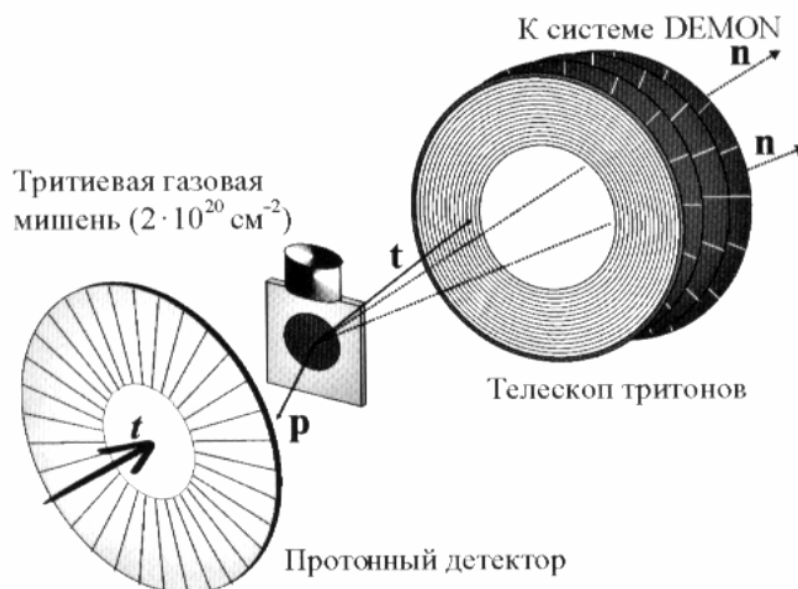


Рис.28. Схема ядерно-физической части эксперимента на установке АКУЛИНА.

В экспериментах пучок ионов трития, ускоренный на циклотроне У-400М, доставляется сепаратором ACCULLINA [10] к тритиевой или дейтериевой мишени, где происходят реакции с образованием нейтронно-избыточных ядер (Рис.28). Всю ядерно-физическую часть обеспечил ОИЯИ. Заряженные частицы регистрирует система из позиционно-чувствительных телескопов на кремниевых E и ΔE -детекторах, нейтронов распада - времяпролетный спектрометр ДЕМООН [79].

Сотрудниками ВНИИЭФ решались две задачи. Первая задача заключалась в обеспечении подачи потока изотопов водорода с заданной концентраций молекул ИВ в высокочастотный ионный источник циклотрона для формирования

третиевого пучка. Вторая состояла в создании уникальной жидкотритиевой мишени (толщина рабочего слоя жидкого трития 0.4 мм, входного/выходного окна из нержавеющей стали - 20 мкм).

Для решения первой задачи была создана автоматизированная система подготовки смеси изотопов водорода (СПИВ) [12,13] и утилизации смеси ИВ по завершении работ. Для решения второй, кроме собственно мишени, был создан автоматизированный комплекс [14,15,16] для заполнения мишени тритием, обеспечения температурного режима её эксплуатации, эвакуации и утилизации трития. Созданные аппаратные комплексы были оснащены системой контроля радиационной обстановки в рабочих комнатах и технологических коммуникациях.

2.2. Управление системой подачи ИВ в ионный источник циклотрона У-400М

СПИВ (Рис.29) состоит из блока подачи ИВ (БПИВ) для подачи молекул ИВ заданного состава в ионный источник и тонкой регулировки их потоков, системы утилизации (СУ) для удаления трития при технологических операциях с газовыми коммуникациями СПИВ и очистки криогенных насосов ионного источника циклотрона. Для управления СПИВ создана автоматизированная СКУ [12].

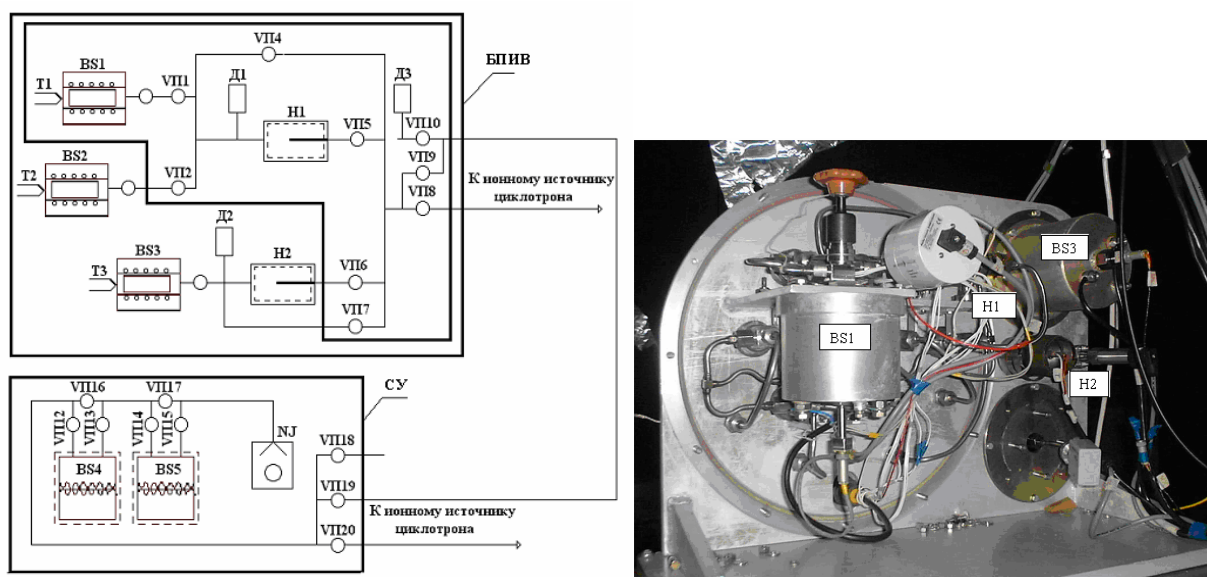


Рис.29. Принципиальная схема и внешний вид СПИВ.

БПИВ – блок подачи ИВ; СУ - система утилизации; BS1-BS3 - источники ИВ; BS4-BS5 – ловушки; H1, H2 - нагреватели; Д1 – измеритель вакуума (PRK261); Д2 - датчик давления (TRK261); Д3 – датчик давления (CMR261); NJ – насос форвакуумный (ВОС EDWARDS GVSP30); VP – вентили сильфонные вакуумные; Т – термопары.

БПИВ включает термодесорбционные источники ИВ BS1-BS3, температуры их нагрева измеряют термопары Т1–Т3. Тритиевый источник BS1, помещенный в

герметичном объеме, используется в рабочем режиме (при ускорении трития), протий-дейтериевый источник BS2 - при настройке тракта циклотрона и сепаратора ACCULLINA, а дейтериевый источник BS3 - при настройке ионного источника циклотрона и в рабочем режиме. Датчики Д1, Д2, Д3 измеряют давление, а форвакуумный насос NJ откачивает газовые коммуникации и объемы.

ИВ, включая тритий, хранятся в химически связанном состоянии на ^{238}U . Равновесное давление P ИВ над гидридом урана зависит от температуры [12]:

$$\begin{aligned} \lg P(\text{Па}) &= -4590/T + 11,59 && \text{для протия} \\ \lg P(\text{Па}) &= -4500/T + 11,56 && \text{для дейтерия} \\ \lg P(\text{Па}) &= -4471/T + 11,73 && \text{для трития} \end{aligned} \quad (31)$$

При комнатной температуре давление трития составляет $\sim 7 \cdot 10^{-4}$ Па. Для заполнения ампулы натекателя ИВ под давлением $\sim 0,1$ МПа достаточен нагрев источников до температуры (680–705) К. Конструкция источника BS1 и герметичного объема СПИВ обеспечивают три рубежа защиты, что в совокупности со свойствами тритида урана позволяет классифицировать источник BS1 как закрытый источник ионизирующего излучения.

Потоки подаваемых в источник ИВ регулируют натекатели Н1 и Н2. Они созданы на основе Ni капилляра, помещенного в ампулу с газом (Рис.30). Поток натекателя определяют: материал и геометрия капилляра, давление ИВ в ампуле и температура нагрева капилляра при пропускании через него электрического тока.

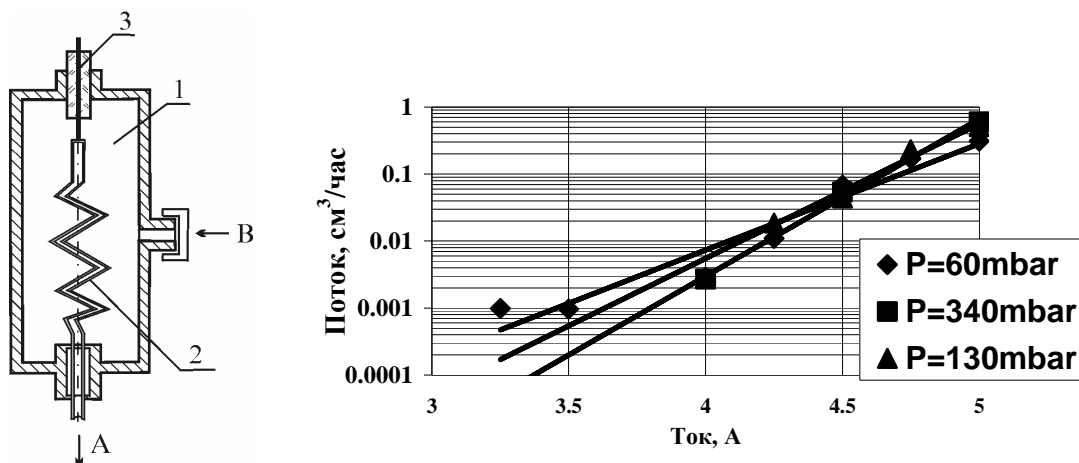


Рис.30. Схема натекателя и его характеристика (зависимость потока ИВ от тока капилляра).

1 – корпус натекателя, 2 – никелевый капилляр, 3 – электровод. А и В – выходной и входной потоки.

СПИВ работает следующим образом. При открытом вентиле источника BS3 и его нагреве до нужной температуры в ампулу натекателя Н2 подается дейтерий. По достижении заданного давления Д2 вентиль источника закрывается, а нагрев отключается. Включается нагрев натекателя Н2 и регулировкой его температуры

задается требуемый поток дейтерия ($0,1 \div 5 \text{ см}^3/\text{час}$). Затем к нему по той же схеме добавляется некоторый ($0,01 \div 0,1 \text{ см}^3/\text{час}$) поток молекул DT из источника BS1. Величины потоков ИВ регулируются током натекателей Н1 и Н2, исходя из требуемой интенсивности ускоренного пучка ионов (DT^+ , HT^+ , HD^+). Вопросы эвакуации и утилизации ИВ по окончании работ описаны в работе [12].

СПИВ находится на циклотроне в опасных для персонала условиях высокого напряжения, магнитного поля и облучения, поэтому ручное управление исключено. По условиям работы ионного источника БПИВ стоит на изолированной платформе прямо над циклотроном под потенциалом $\sim 15 \text{ кВ}$, его электронная аппаратура - в изолированной стойке под тем же потенциалом, а аппаратура СУ трития - в заземленной стойке. Кроме того, при работе циклотрона на аппаратуру АСКУ действует сильное магнитное поле ускорителя и мощные помехи от ВЧ генератора ионного источника. Поэтому при создании АСКУ особое внимание уделялось гальванической изоляции входных и выходных цепей, помехозащитности измерительных и управляющих каналов.

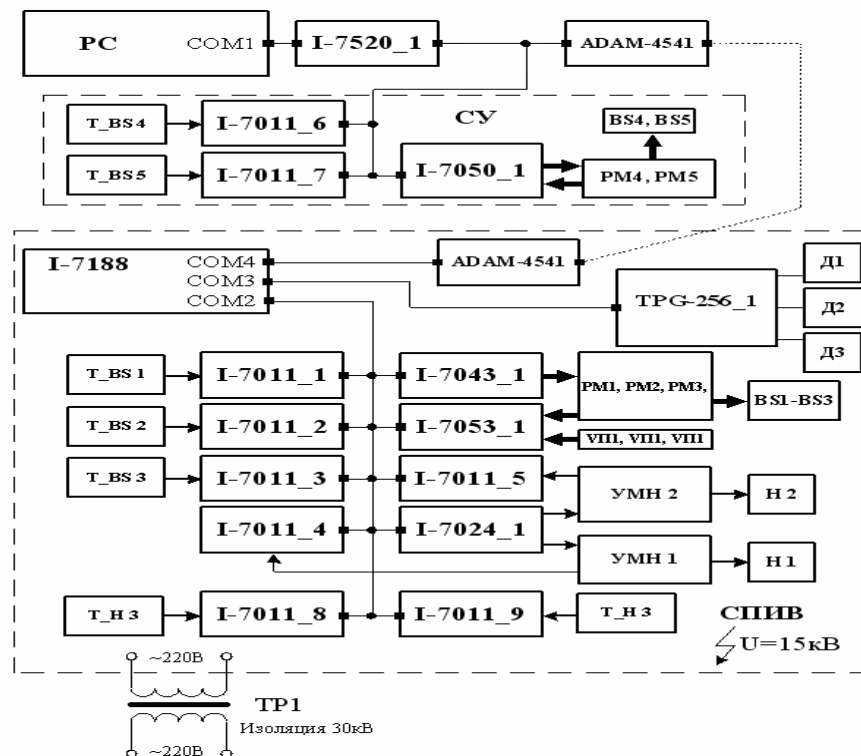


Рис.31. Схема АСКУ для системы подачи ИВ.

АСКУ предназначена для дистанционного (с пульта циклотрона) управления всеми узлами СПИВ, тонкого регулирования потоков молекул ИВ за счет регулируемого нагрева натекателей и стабилизации их температуры, графического

и цифрового отображения контролируемых процессов и ведения протокола эксперимента с записью данных на жесткий диск. Она представляет собой распределенную сеть, состоящую из управляющего компьютера в пультовой комнате на расстоянии ~100 м от циклотрона, а также набора модулей аналогового и дискретного ввода/вывода серии I-7000, датчиков вакуума с контроллером TRG-256 фирмы Balzers. В состав СКУ входит 29 измерительных каналов (14 аналоговых, 15 цифровых), из них 8 каналов для управления (Таблица 4).

Таблица 4. Список измерительных каналов АСКУ АКУЛИНА.

Измерение и управление	Каналов	Тип канала
АКУЛИНА – СИСТЕМА ПОДАЧИ ИЗОТОПОВ ВОДОРОДА		
Термопарные измерения температуры нагревателей источников	3	аналоговый
Управление нагревателями источников	3	дискретный
Контроль обрыва спиралей нагревателей источников	3	дискретный
Управление релейной блокировкой нагревателей источников	3	дискретный
Измерение температуры холодного спая (окружающей среды)	1	аналоговый
Измерение низких давлений датчиками вакуума фирмы Balzers	4	RS-232
Измерение тока и напряжения натекателей	4	аналоговый
Управление током натекателей	2	аналоговый
Контроль состояний ручных вентилях с датчиками положения	6	дискретный
АКУЛИНА – ТРИТИЕВАЯ МИШЕНЬ		
Термопарные измерения температуры нагревателей источников	5	аналоговый
Управление нагревателями источников	5	дискретный
Контроль обрыва спиралей нагревателей источников	5	дискретный
Управление релейной блокировкой нагревателей источников	5	дискретный
Измерение температуры холодного спая (окружающей среды)	1	аналоговый
Измерение температуры мишени криогенным термодиодом	1	аналоговый
Стабилизация температуры мишени	1	аналоговый
Измерение низких давлений датчиками вакуума фирмы Balzers	4	RS-232
Контроль состояний ручных вентилях с датчиками положения	34	дискретный
Управление электромагнитными клапанами	6	дискретный
Контроль состояний электромагнитных клапанов	6	дискретный
Контроль состояний электроконтактных мановакуумметров	2	дискретный
Контроль объемной активности трития в рабочем помещении	5	RS-485

В составе АСКУ СПИВ - две подсистемы: первая обслуживает БПИВ, а вторая - систему утилизации СУ (Рис.31). Для гальванической развязки БПИВ по питанию служит трансформатор с изоляцией ~30 кВ, а по каналу обмена данными - оптоволоконная линия связи на модулях ADAM-4051 фирмы Advantech.

АСКУ подсистемы СПИВ стабилизирует температуру нагрева источников ИВ BS1÷BS3 в диапазоне 20÷600°C в пределах $\pm(1\div2)^\circ\text{C}$ методом ШИМ (раздел 3.3.2). ПК программно формирует сигнал ШИМ и подает его на электронные регуляторы PM1÷PM3 через многоканальные модули дискретного ввода-вывода типа I-7043. Температуру нагрева измеряют термопары типа хромель/копель T_BS1÷T_BS3, их термоЭДС регистрируют модули АЦП I-7011 с гальванической изоляцией ~3 кВ. Использование многоканальных модулей АЦП I-7018 оказалось невозможным, т.к. из-за электромагнитных наводок скачки синфазного напряжения между входами превышали допустимую для этих модулей величину 35 В.

Вакуум в газовых коммуникациях системы подготовки и герметичном объеме измеряется датчиками PKR261, TRK261, CMR261 (Д1-Д3) фирмы Balzers, подключенными к 6-канальному контроллеру TPG-256.

Потоки ИВ регулируются пропусканием тока 0÷6А через никелевую трубку натекателя, включенную в цепь нагрузки усилителя мощности УМН. Управляющее напряжение на вход УМН подается через гальванически изолированный 14-разрядный модуль ЦАП I-7024. Для контроля тока в выходную цепь усилителя включено сопротивление 0,05 Ом, падение напряжения на котором регистрируется модулем АЦП I-7011. Для подавления ВЧ помех на входе токового усилителя установлены фильтры. Термопары T_H1, T_H2, подключенные к модулю АЦП I-7011, измеряют температуры натекателей Н1, Н2. Многоканальный модуль дискретного ввода I-7053 контролирует состояния вентилях ВП1, ВП2, ВП4.

АСКУ системы утилизации включает 2 канала автоматизированного управления для поглотителей BS4, BS5, идентичных каналам управления источниками BS1-BS3 (термопарные датчики, регуляторы мощности, см. 3.3.2).

Устройства с разными интерфейсами и скоростями передачи (модули I-7000, контроллер TPG-256) в единую сеть объединяет интеллектуальный, адресуемый преобразователь интерфейса на базе микропроцессорного контроллера I-7188, имеющий 4 порта последовательного ввода/вывода. Не адресуемый контроллер

TPG-256 подключен к порту COM3 (RS-232) и, с точки зрения сети RS-485, имеет свой виртуальный адрес. Модули серии I-7000 подключены к порту COM3 (RS-485). Порт COM4 используется для выхода на управляющий компьютер.

ПО АСКУ системы подачи ИВ разработано в среде пакета CRW-DAQ в виде 19 параллельно работающих программных устройств (9 драйверов аппаратуры, 10 прикладных программ) на языке DAQ Pascal с периодом опроса от 50 до 250 мс.

Система подачи ИВ управляется через главную мнемосхему (Рис.32), близкую по виду к газовой схеме установки и отображающую в реальном времени состояния узлов газовой системы (вентилей, нагревателей, натекателей, насосов) в привычном для оператора виде, что облегчает его работу.

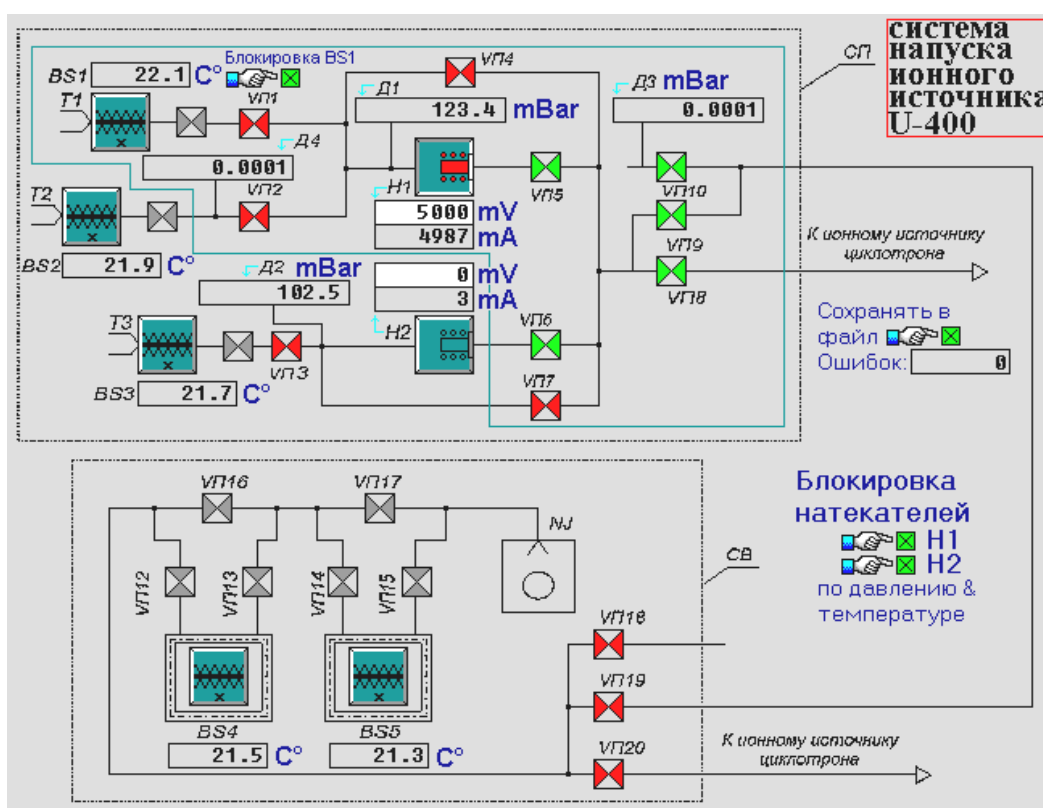


Рис.32. Главная мнемосхема для управления СПИВ.

АСКУ блокирует натекатели Н1, Н2 по давлению и температуре при неожиданном нарушении нормальных режимов работы, предохраняя их поломки. Блокировка нагревателя BS1 по давлению, температуре и состоянию вентиля предохраняет источник трития от опасных режимов работы.

2.3. Управление комплексом тритиевой мишени

Комплекс тритиевой мишени [14], внешний вид которого представлен на рисунке (Рис.33), состоит из четырех подсистем (Рис.34).



Рис.33. Внешний вид газового комплекса тритиевой мишени установки АКУЛИНА.

Мишенный блок (МБ) расположен во внутренней полости реакционной камеры (РК) сепаратора АКУЛИНА [10] и обеспечивает радиационно-безопасное хранение трития в процессе эксперимента в жидком или газообразном состоянии при заданной температуре. Система наполнения (СН) обеспечивает подачу в мишень ИВ или гелия и позволяет многократно использовать один и тот же изотоп при смене газа в мишени. Система утилизации (СУ) обеспечивает эвакуацию и безопасное хранение трития. Комплекс работает под управлением АСКУ [16].

Основной элемент комплекса МБ - тритиевая мишень (ТМ). Это - сосуд с тонкостенными (12,5 мкм) окнами и двумя рубежами защиты. ТМ (Рис.34,а) содержит герметичную цилиндрическую ячейку С, ограниченную с двух сторон тонкими мембранами W1, W2 из нержавеющей стали, в которую из СН подается газ через трубку Т1 (путь подачи и эвакуации газа GSE). Стенки ячейки С образуют первый защитный барьер от утечки трития в реакционную камеру. Объемы P1, P2, образованные внешними окнами WP1, WP2 и соединенные трубками Т2, Т3 с геттером G1 (порошкообразный титан), образуют второй барьер. При утечке через окна W1, W2 или трубку Т1 тритий будет полностью поглощен геттером [14,15,16].

ТМ заполняется тритием в реакционной камере, являющейся третьим рубежом защиты от несанкционированного проникновения трития в окружающую среду.

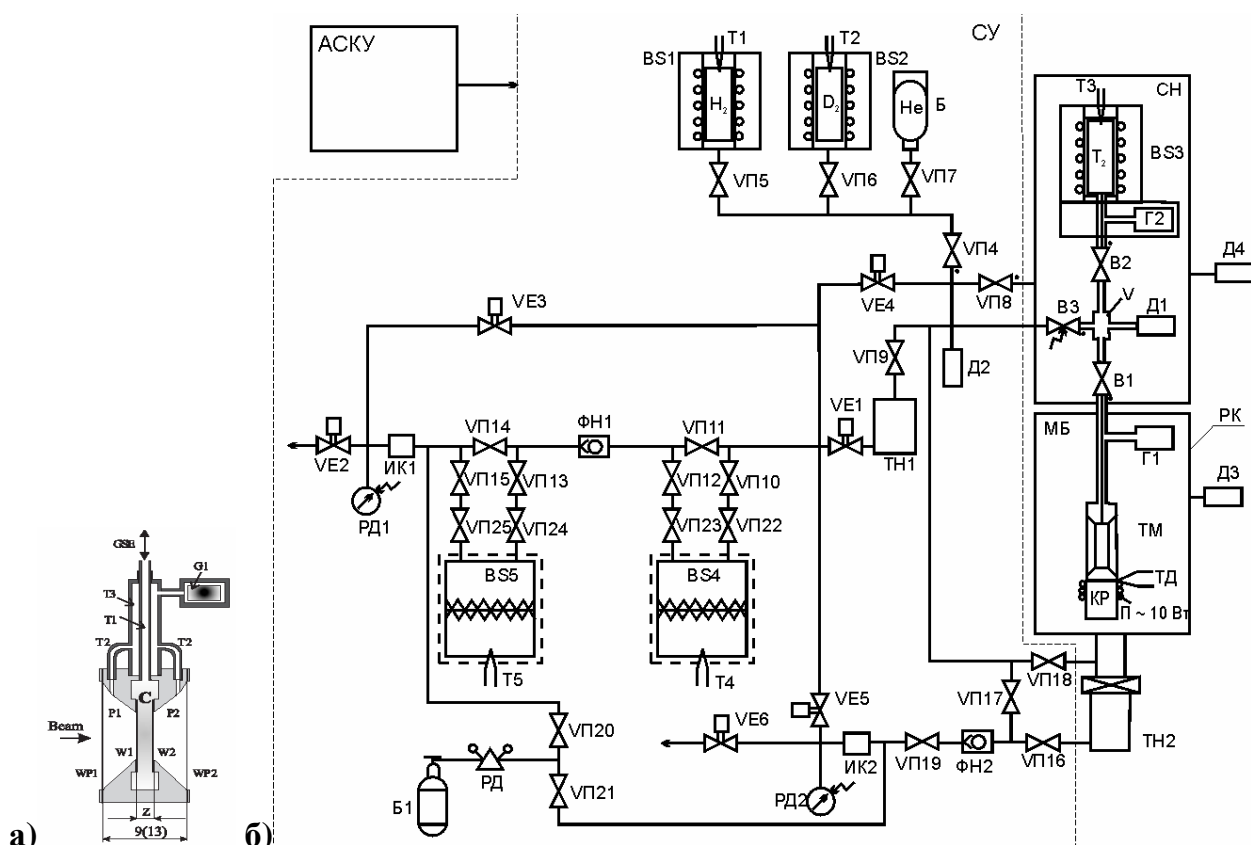


Рис.34. Принципиальная схема тритиевой мишени (а) и комплекса её газового обеспечения.

АСКУ – автоматизированная система контроля управления; СН – система наполнения; СУ – системы утилизации трития и контроля радиационной обстановки; МБ – мишенный блок; ТМ – тритиевая мишень; РК – реакционная камера; BS1/2/3 – источник протия /дейтерия/трития; BS4(5) – ловушка; Б(Б1) – баллон с гелием; Д1(4) – измеритель вакуума (PRK261); Д2 – датчик давления (TRK261); Д3 – датчик давления (CMR261); ИК1(2) – ионизационная камера; РД1(2) – электроконтактный мановакуумметр (ДА2005); ФН1(2) – насос форвакуумный (BOC EDWARDS GVSP30); ТН1(2) – насос турбомолекулярный (STP – 300M); В1-3, VP4-21 – вентили; VP22-25 – вентили сильфонные вакуумные; VE1-6 – вентили электромагнитные; V – мерная емкость; Т1-Т5 – термopара; Г1(2) – геттер; ТД – термодиод; КР – криорефрижератор GOOL POWER 2/10

Для заполнения мишени соответствующий источник ИВ соединяется с заранее откакумированным измерительным объемом V (Рис.34,б) системы наполнения и нагревается до температуры, дающей нужное давление заданного ИВ по формуле (31). После заполнения объема V нагрев отключается, а газ запускается в мишень. При охлаждении источников идет обратное поглощение ИВ из газовых коммуникаций, что позволяет многократно использовать один и тот же газ, а также максимально уменьшить количество трития, уходящего в систему утилизации.

При работе жидкого и газового вариантов ТМ температура ячейки должна составлять $18 \div 21$ К и $23 \div 30$ К соответственно. Охлаждение системы обеспечивает криорефрижератор (КР) типа Cool Power 2/10 (Leybold), установленный на крышке

РК. Перед охлаждением газа в мишени внутренняя полость РК вакуумируется до $10^{-7} \div 10^{-8}$ Па. Система наполнения, заполненная рабочим газом, через газовую магистраль соединяется с внутренним объемом мишени. Затем включается криорефрижератор и начинается охлаждение, контролируемое по показаниям термодиода ТД. При определении температуры газа в мишени принимается, что в установившемся режиме она равна температуре корпуса мишени.

Тритий активностью до 1кКи ($3.7 \cdot 10^{13}$ Бк) хранится и транспортируется в химически связанном состоянии на ^{238}U в источнике BS3, входящем в состав СН. Равновесное давление ИВ для урана при комнатной температуре равно $\leq 10^{-3}$ Па, а температура его разложения по формуле (31), при которой давление ИВ становится выше атмосферного, составляет 680÷705К. Конструкция BS3 и СН с тремя рубежами защиты, вместе со свойствами тритида урана позволяют классифицировать BS3 как закрытый источник ионизирующего излучения.

В собранном виде ТМ и источник трития BS3 представляют собой замкнутую систему, позволяющую удерживать тритий в жидком или газообразном виде в процессе эксперимента, либо переводить его в химически связанное состояние для длительного хранения в источнике. Ряд технологических операций нарушает замкнутость этой системы. К ним относятся: заполнение мишени ИВ или гелием, требующееся по условиям физического эксперимента; удаление из мишени радиогенного гелия, образующегося в результате распада трития при длительной работе; разгерметизация газовых коммуникаций при смене или демонтаже мишени.

Вопросы очистки газовых коммуникаций, контактировавших с тритием, перед проведением всех этих операций и технология утилизации ИВ подробно описана в работе [14]. Важно отметить, что при управлении процессом утилизации используются результаты измерений ОА трития камерой ИК1. Если показания ИК1 превышают допустимые нормы, газовый поток направляется на ловушку BS5 для дополнительной очистки. Если даже после этого показания ИК1 превышают допустимые пределы, происходит автоматическая блокировка вентиля VE2, открывается вентиль VE3 и установка утилизации переводится в режим рециклинга. При этом газовая смесь прокачивается через обе ловушки до тех пор, пока показания ИК1 не станут ниже допустимых норм.

Для управления узлами комплекса тритиевой мишени, контроля и поддержания их параметров, заполнения мишени тритием, регулирования и стабилизации температуры мишени, контроля объемной активности трития в газовых коммуникациях системы и рабочих помещениях создана автоматизированная система контроля и управления [16]. Она обеспечивает также графическое и цифровое отображение контролируемых процессов, дистанционное наблюдение и ведение протокола эксперимента с записью данных на жесткий диск.

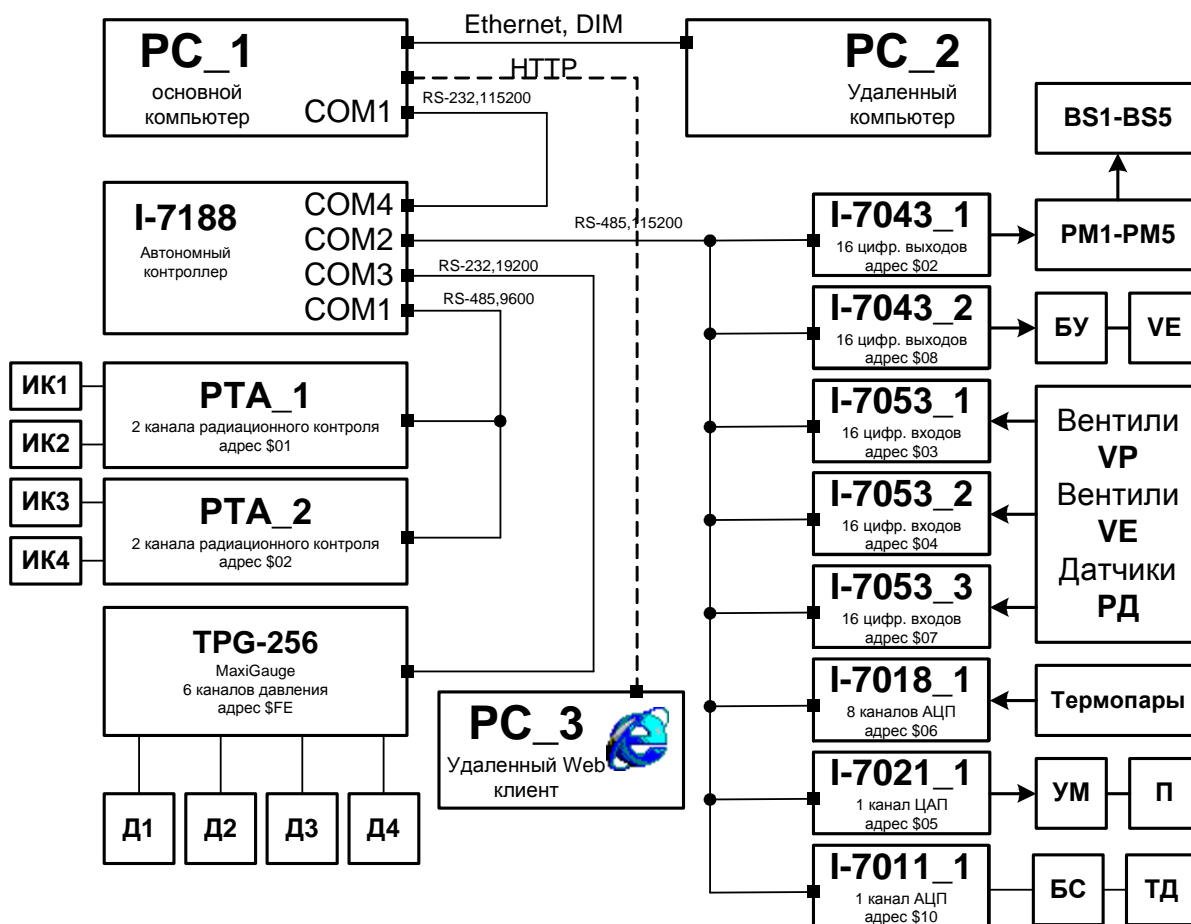


Рис.35. Блок-схема АСКУ комплекса тритиевой мишени.

АСКУ комплекса (Рис.35) представляет собой распределенную сеть, состоящую из 2 компьютеров и набора автономных сетевых модулей, связанных между собой по интерфейсу RS-232 и RS-485. Она включает 80 измерительных каналов (17 аналоговых, 63 цифровых), из них 17 каналов управления (Таблица 4).

В сеть включены разнотипные устройства с разными скоростями передачи. Микропроцессорный контроллер I-7188 используется как интеллектуальный, адресуемый преобразователь интерфейса. К портам контроллера COM1, COM2, COM3 подключаются модули серии I-7000, контроллер TPG-256 и радиометры

РТА_1, РТА_2. Порт COM4 обеспечивает подключение к стандартному последовательному порту центрального управляющего компьютера PC_1, служащего для сбора данных и активного управления комплексом.

Все регистрируемые параметры через сеть Ethernet по протоколу DIM [70] (Приложение 7) передаются на удаленный компьютер PC_2, расположенный в пультовой циклотрона и обеспечивающий прием всех регистрируемых параметров от PC_1 и отображение состояния комплекса.

В состав АСКУ входит встроенный Web сервер (Приложение 7,Рис.70), поэтому наблюдение за состоянием установки возможно и по протоколу HTTP, через удаленного Web клиента PC_3. Удаленное наблюдение необходимо из-за повышения радиоактивного фона в зоне мишени, где установлен компьютер PC_1, во время рабочих измерений, поэтому пребывание персонала там нежелательно.

Для источников изотопов водорода BS1-BS3 и ловушек BS4-BS5 создана система управления нагревателями, аналогичная описанной в разделе 3.3.2. Она использует метод ШИМ, полупроводниковые регуляторы мощности РМ с управлением через модули дискретного вывода I-7043, термопарные измерения многоканальным модулем АЦП I-7018. Температура стабилизируется с точностью около $\pm(1\div 2)^\circ\text{C}$, при превышении предельных значений заданных параметров питание нагревателей автоматически отключается. Вакуум измеряют датчики Д1-Д4 фирмы Balzers через 6-канальный контроллер TPG-256, включенный в локальную сеть АСКУ двухпроводной линией связи по интерфейсу RS-485.

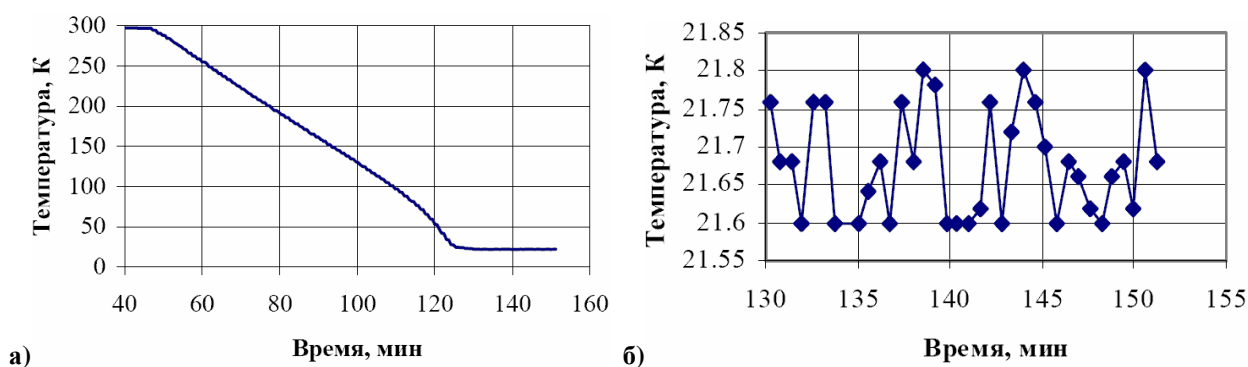


Рис.36 Характерный вид кривой охлаждения (а) и температура мишени на участке стабилизации (б).

АСКУ комплекса обеспечивает контроль за состоянием вентилях VP; контроль и управление состоянием электромагнитных клапанов VE1÷VE6 (через блок управления БУ); контроль состояний концевых выключателей мановакууметров РД1, РД2; блокировку выброса газа в вентиляцию при

превышении заданного уровня объемной активности газа и выполняет ряд других функций, обеспечивающих безопасность проведения работ.

Для измерения температуры мишени (Рис.34) используется кремниевый диод (ТД), подключенный через электронный блок согласования (БС), включающий стабилизатор тока на 10 мкА и буферный дифференциальный усилитель с входным сопротивлением 100 Мом и коэффициентом передачи, равным 1. Напряжение на выходе усилителя измеряется одноканальным модулем АЦП I-7011 с точностью 0,05%. Диапазон измерений температуры (1,4÷330)К, точность измерения в диапазоне (2÷100)К составляет ±0,1К, а в диапазоне выше 100К - около ±1%.

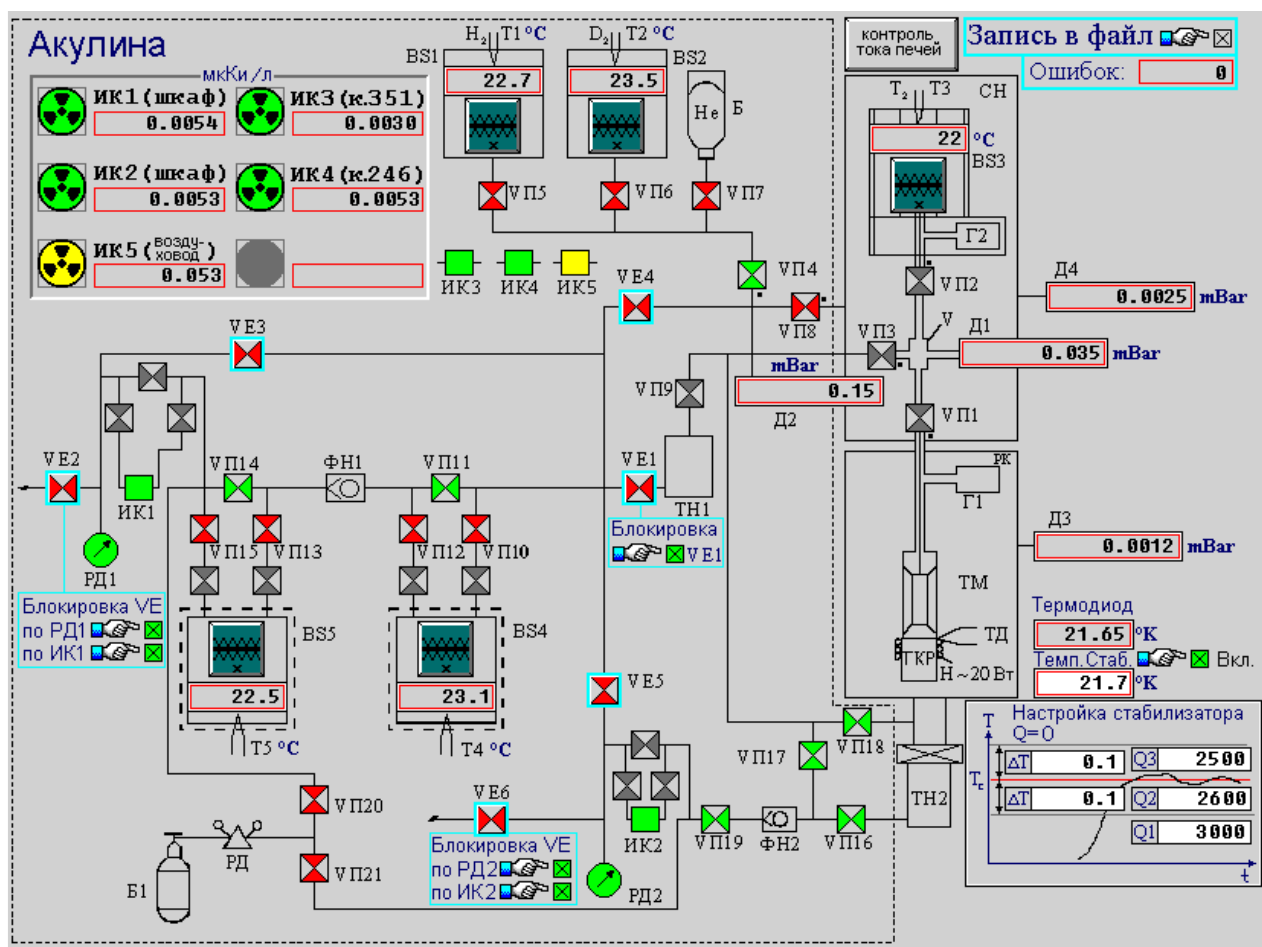


Рис.37. Главная мнемосхема комплекса тритиевой мишени установки АКУЛИНА.

АСКУ обеспечивает охлаждение до температуры 18÷21 К для жидкой и 23÷30 К для газовой мишени. При этом криорефрижератор постоянно включен, а регулирование и стабилизацию температуры мишени выполняет нагреватель (П) с мощностью до 20 Вт. Нагреватель управляется модулем ЦАП I-7021 через усилитель мощности (УМ). После охлаждения мишени до требуемой температуры

(Рис.36,а) включается система термостабилизации и температура поддерживается нагревателем Н с точностью не хуже $\pm 0,1$ К в течение всего опыта (Рис.36,б).

Отображение состояния комплекса мишени и управление исполнительными устройствами газовой системы, а также задание параметров стабилизации температуры мишени осуществляется через главную мнемосхему (Рис.37). Она имеет близкий к газовой схеме комплекса вид и отображает в реальном времени состояния узлов газовой системы (вентилей, нагревателей, насосов) в привычном для пользователя виде, что облегчает работу оператора. Мнемосхема позволяет вызывать графики изменения интересующих величин, управлять клапанами, задавать параметры работы нагревателей.

ПО АСКУ комплекса мишени реализовано в виде 25 параллельно работающих программных устройств (11 драйверов аппаратуры, 14 прикладных программ) на языке DAQ Pascal с периодом опроса от 10 до 1000 мс.

АСКУ включает подсистему дозиметрического контроля по тритию, обеспечивающую мониторинг ОА трития в газовой системе установки (ионизационные камеры ИК1 и ИК2) и контроль ОА трития в воздухе рабочей зоны (ИК3, ИК4). Измерения ОА осуществляются двумя радиометрами РТА-4. Предусмотрено звуковое предупреждение персонала при превышении заданных порогов ОА. Управление системой дозконтроля осуществляется через программное обеспечение и мнемосхемы, аналогичные описанным в разделе 3.3.5, Рис.55.

2.4. Результаты экспериментов на установке АКУЛИНА

С использованием автоматизированной системы подачи ИВ в ионный источник циклотрона У-400М и автоматизированного комплекса тритиевой мишени на сепараторе АКУЛИНА циклотрона У-400М был выполнен ряд экспериментов по изучению резонансных уровней ядер ^4H и ^5H [34,35,36,37,38].

Потоки ИВ, подаваемые в ионный источник, регулировались дистанционно из пультовой циклотрона с помощью автоматизированной системы подачи ИВ. Потоки D и T составляли 1,7 и 0,06–0,1 см³/ч. Компоненты тока ионов D_2^+ , D_3^+ и D^+ , выходящего из источника (Рис.38), имели величину около 30, 10 и 6 мкА.

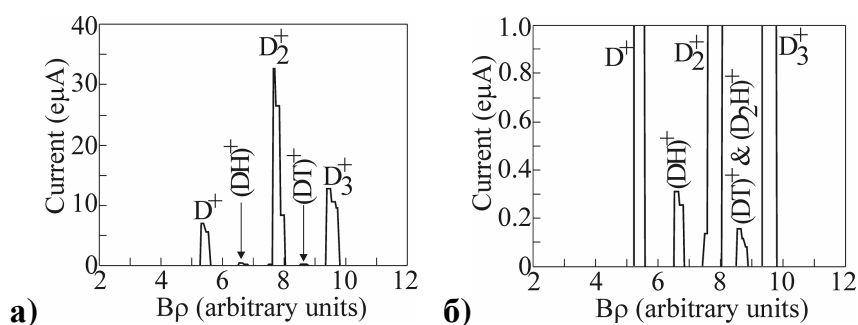


Рис.38. Спектр ионов D/T [13]: полная шкала (а), увеличенная шкала (б).

Суммарный ток ионов DT^+ и D_2H^+ был около 100 нА. Ток пучка ускоренных тритонов с энергией 58,2 МэВ, выведенного из циклотрона У-400М методом обдирки на тонкой графитовой фольге, был около 10 нА. Интенсивность пучка тритонов на входе в сепаратор составляла $1 \cdot 10^9 \text{ c}^{-1}$. Коллимированный пучок тритонов интенсивностью $3 \cdot 10^7 \text{ c}^{-1}$ фокусировался в пятно $\varnothing 5 \text{ мм}$ на мишени, помещенной в центре реакционной камеры. Энергия тритонов в тритиевой мишени равнялась 57,5 МэВ, энергетический разброс пучка не превышал 330 кэВ.

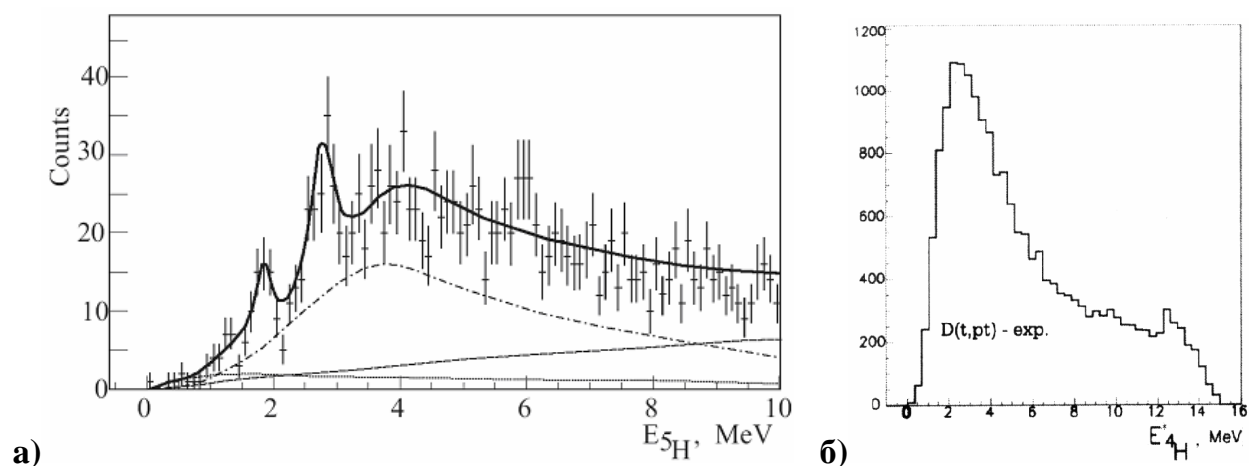


Рис.39. Спектры энергий ^5H в реакции $t(t, pt)nn$ (б) и ^4H в реакции $d(t, pt)n$ (а), полученные методом недостающей массы.

Для синтеза тяжелых ИВ (^4H , ^5H) в реакциях передачи нейтронов $t+t \rightarrow p+^5\text{H}$, $t+t \rightarrow d+^4\text{H}$, $t+d \rightarrow p+^4\text{H}$ использовались тритиевые и дейтериевые мишени. Мишенный комплекс обеспечивал заправку мишеней и стабилизацию их температуры в диапазоне $18 \div 300\text{K}$ с точностью $\pm 0,1\text{K}$ (Рис.36,б). В измерениях с тритиевым пучком и жидкой тритиевой мишенью были исследованы резонансные состояния ядра ^5H в реакции передачи двух нейтронов $t+t \rightarrow p+^5\text{H}$ [34,35,36]. При анализе тройных совпадений $p+t+n$ в спектре энергий ^5H обнаружен резонанс с энергией $(1,8 \pm 0,1) \text{ МэВ}$ (Рис.39,а) с неожиданно малой наблюдаемой шириной

$\Gamma_{\text{набл}} \leq 0,5$ МэВ. В работе [36] наличие этого пика было объяснено интерференцией основного (спин $1/2^+$) и возбужденных состояний ($3/2^+$ и $5/2^+$) ядра ${}^5\text{H}$.

В измерениях с тритиевым пучком и жидкой дейтериевой мишенью в реакции $d+t \rightarrow p+{}^4\text{H}$ были изучены резонансные уровни ядер ${}^4\text{H}$. В спектре энергии ядра ${}^4\text{H}$ (Рис.39,б) наблюдается резонанс с энергией $E_{\text{рез}} = (3,22 \pm 0,15)$ МэВ и наблюдаемой шириной $\Gamma_{\text{набл}} = (3,33 \pm 0,25)$ МэВ. Результаты этих экспериментов обобщены в диссертации С.И. Сидорчука [39].

2.5. Выводы

Для экспериментов по получению и изучению нейтронно-избыточных ядер ${}^4\text{H}$ и ${}^5\text{H}$, проводящихся в ЛЯР ОИЯИ, созданы автоматизированные системы для управления напуском изотопов водорода в ионный источник циклотрона У-400М [12,13] и управления газовым комплексом жидкой тритиевой мишени [14,15,16].

Система управления подачей изотопов водорода в ионный источник обеспечивает тонкое регулирование потоков ИВ и точное поддержание рабочих параметров пучка тритонов, несмотря на работу в жестких условиях функционирования электронной аппаратуры на ускорителе [12,13]. АСКУ комплекса тритиевой мишени обеспечивает управление процессами заполнения мишени и утилизации трития, стабилизацию температуры мишени в ходе экспериментов, а также радиационный контроль объемной активности трития в газовых коммуникациях комплекса и рабочих помещениях [14,15,16].

С использованием разработанных автоматизированных систем проведены эксперименты по изучению резонансных состояний ядер ${}^4\text{H}$ и ${}^5\text{H}$, образующихся при облучении дейтериевой или тритиевой мишени пучком тритонов, в которых получены новые физические результаты [34,35,36,37,38,39].

В многочисленных экспериментах автоматизированные системы подачи ИВ и комплекса тритиевой мишени работали стабильно и безотказно, обеспечив безопасность работ с тритием, поддержание условий эксперимента, измерение ряда ключевых физических параметров (потока ИВ в источнике, температуры мишени).

3. Автоматизация экспериментов на исследовательском стенде ПРОМЕТЕЙ

3.1. Исследовательский стенд ПРОМЕТЕЙ

Стенд ПРОМЕТЕЙ (Рис.40) был создан в РФЯЦ-ВНИИЭФ [18] для изучения явлений накопления и пропускания трития металлами и КМ в 2000-м году. Он состоит из двух объединенных исследовательских комплексов, решающих две научные задачи. Первая - задача исследования явления сверхпроницаемости трития сквозь металлические мембраны, вторая - изучение проницаемости, проникновения и накопления трития конструкционными материалами, поиск и исследование защитных покрытий, повышающих безопасность использования тритий-содержащих газовых сред [40,41,42,43,44,45,46,71,72,73,74].

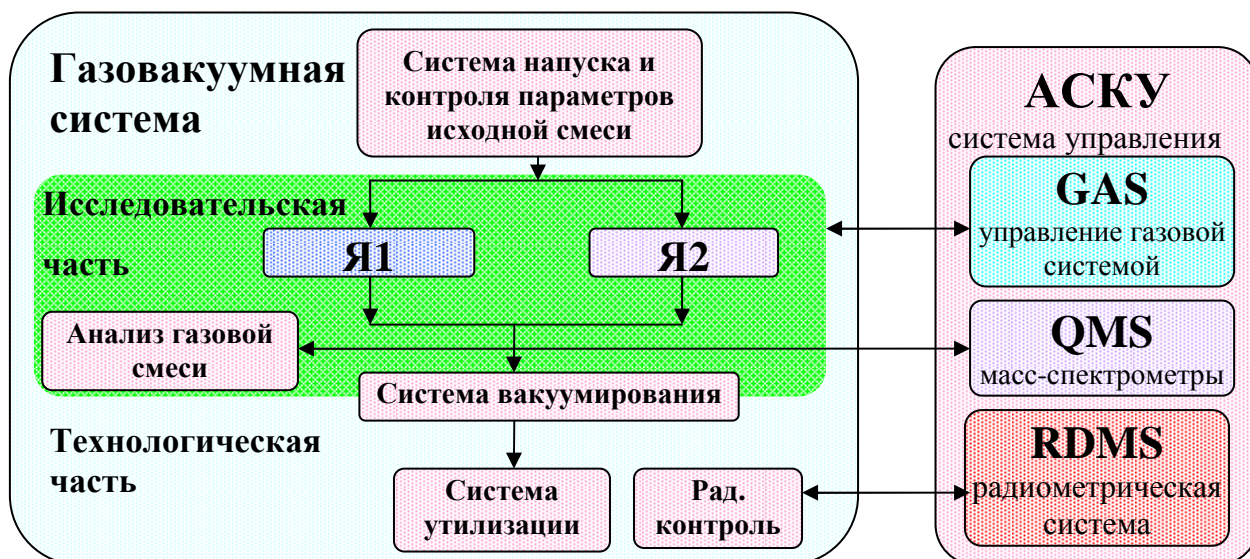


Рис.40. Структурная схема стенда ПРОМЕТЕЙ.

Стенд создан для опытов со смесями ИВ при давлениях ниже атмосферного. Его оборудование и газовая схема (Рис.41) подробно описаны в работе [17]. Структурно стенд состоит из газовакуумной, масс-спектрометрической, радиометрической систем и АСКУ. Газовакуумная система делится на две части – исследовательскую, для проведения основных физических измерений и технологическую, для подготовки и утилизации газовой смеси ИВ.

В исследовательскую часть входят ячейки Я1 и Я2. Ячейка Я1 служит для исследования защитных покрытий металлов и КМ методом концентрационных импульсов (МКИ) в условиях воздействия надтепловых частиц ИВ.

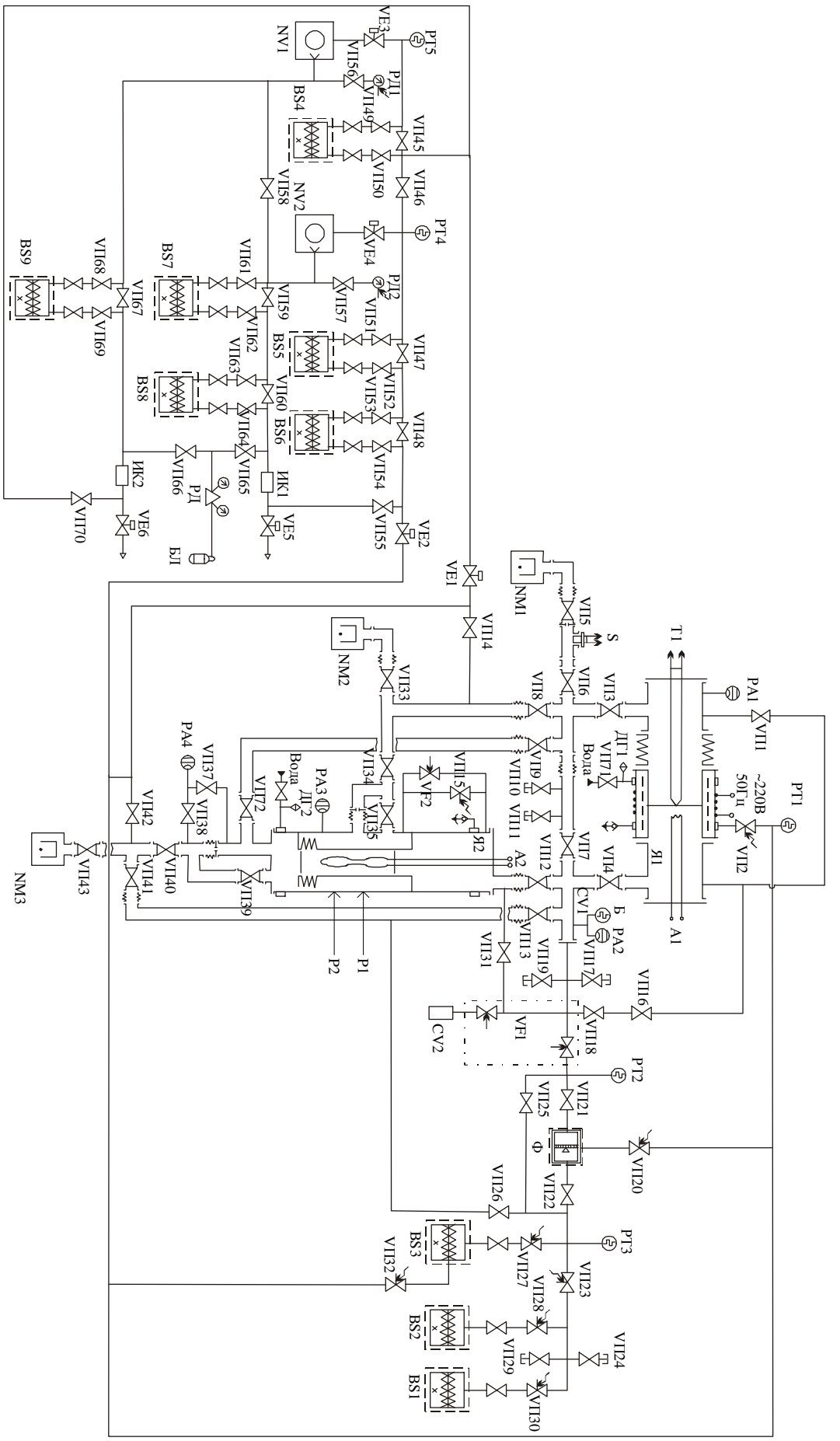


Рис.41. Газовакуумная схема установки ПРОМЕТЕЙ [17].

BS1-BS3 - генераторы изотопов водорода; BS4-BS9 - абсорберы; Ф - фильтр; А1, А2 - автоматизаторы; ВП - вентили; VE - клапаны электромагнитные; VF1-VF3 - регуляторы потока; Я1, Я2 - исследовательские ячейки; CV1, CV2 - емкости; PT - преобразователи термомпарные; PA - преобразователи ионизационные; Б - баратрон; PД - мановакуумметр; ИК - ионизационные камеры; Т1 - термомпара; P1, P2 - пирометры; ДП - датчики давления воды; БЛ - баллон; NM - насосы магнитоэлектрические; NV - насосы форвакуумные;

Ячейка Я2 служит для изучения сверхпроницаемости ИВ на цилиндрических мембранах относительно большой площади, а также исследования и выбора защитных покрытий металлов и КМ на плоских мембранах. К исследовательской части относится также система масс-спектрометрического анализа газовой смеси. Технологическая часть состоит из трех систем - напуска и контроля параметров исходной газовой смеси, вакуумной откачки и системы утилизации трития.

Стенд - это сложная установка, требующая высокого уровня автоматизации контроля и управления. АСКУ стенда должна обеспечивать регистрацию и архивирование большого числа технологических и физических параметров, автоматизацию управления исполнительными устройствами установки, включая газовакуумную (GAS) и радиометрическую (RDMS, RaDioMetry System) систему, а также масс-спектрометрические измерения (QMS, Quadruple Mass Spectrometer). В её задачи входит также обеспечение безопасности при работе с тритием.

Тритий в установке хранится в закрытых источниках активностью до 1 кКи в связанном состоянии, в виде тритида урана ^{238}U . При проведении исследований тритий выделяется из источников путем их нагрева до температуры $650\div 700\text{ K}$ и поступает в газовые коммуникации. При работе в газовых коммуникациях в свободном состоянии может находиться до 10 Ки ($3.7 \cdot 10^{11}$ Бк) трития [40].

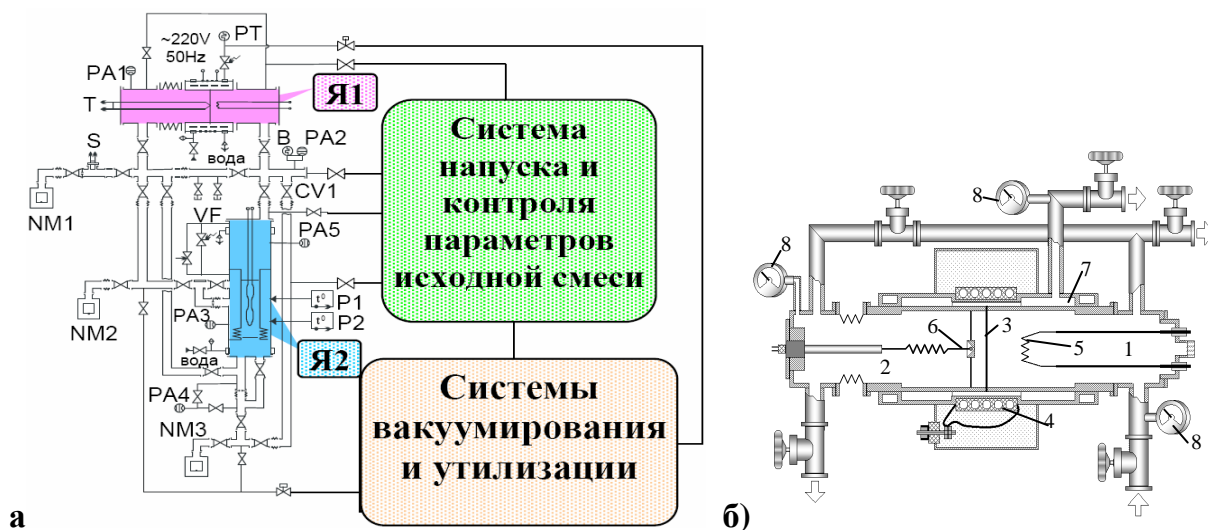


Рис.42. а) Схема исследовательской части газовакуумной системы. б) Ячейка Я1 для изучения проницаемости изотопов водорода через конструкционные материалы.

Исследовательская часть газовой системы

Смесь ИВ (Рис.42,а), подготовленная системой напуска, поступает в измерительные ячейки из емкости CV1. Натекатель VF служит для изучения рециркуляции в Я2. Магниторазрядные насосы NM1, NM2, NM3 откачивают газ на

выходе ячеек и вакуумируют систему. Термопара Т регистрирует температуру мембраны Я1, а масс-спектрометр S – её проницаемость. Датчики PA1...PA5 и PT измеряют давление газа, а пирометры P1,P2 - температуры атомизаторов Я2.

Ячейка Я1 для изучения проникновения и накопления ИВ в металлах

Измерительная ячейка Я1 (Рис.42,б) была разработана в НИИФ СПбГУ, в лаборатории проф. А.А. Курдюмова [40]. Она включает входной (1) и выходной объем (2), разделенные исследуемой плоской мембраной диаметром до 50 мм (3) с электронагревателем (4), для нагрева до 1000К. Температуру мембраны измеряет термопара (6). Насос NM2 непрерывно вакуумирует полость (7) для сбора диффундирующего трития. Давления во входном и выходном объемах измеряют датчики (8). Во входной объем из емкости CV1 подаются ИВ под давлением от $2 \cdot 10^{-7}$ до 1 Па. Выходной объем откачивается насосом NM1. Атомизатор (5) мощностью ~1 Вт диссоциирует молекулы ИВ в атомарное состояние с входной стороны ячейки. Проникающий поток ИВ измеряется масс-спектрометром.

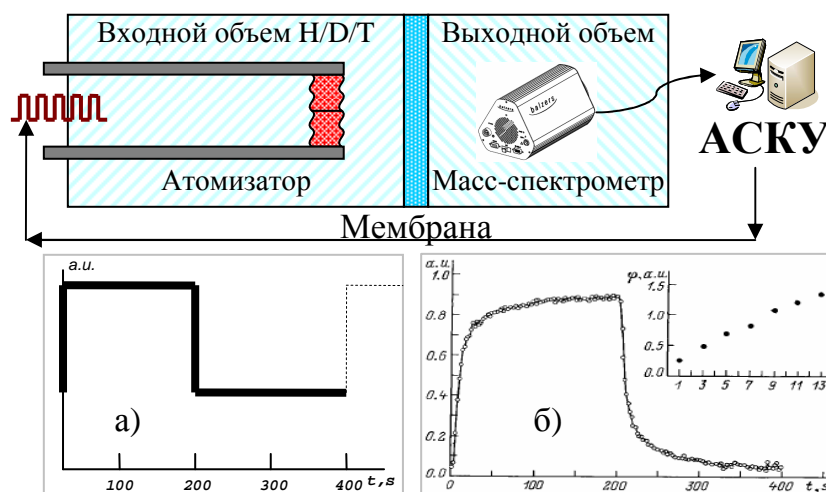


Рис.43. Идея МКИ. а) Входной меандр: периодические прямоугольные импульсы. б) Выходной меандр: усредненный по периодам отклик концентрации и его ФЧХ.

Для исследований на ячейке Я1 используется несколько методов. Наиболее комплексный из них - метод концентрационных импульсов (МКИ), предложенный А.А. Курдюмовым и И.Е. Габисом [74]. Он состоит в следующем (Рис.43). Во входном объеме ячейки с водородом атомизатором создается концентрационный меандр $c_{in}(t)$ - серия прямоугольных импульсов концентрации атомарного водорода с периодом $T \sim 200 \div 400$ секунд. В выходном объеме ячейки масс-спектрометр регистрирует концентрационный отклик $c_{МКИ}(t)$, пропорциональный потоку водорода, проникающего сквозь мембрану. Разложим их в ряд Фурье:

$$c_{in}(t) = \sum_{j=0}^{+\infty} A_j \exp(i\omega_0 j), \quad c_{MKI}(t) = \sum_{j=0}^{+\infty} B_j \exp(i\omega_0 j), \quad \omega_0 = \frac{2\pi}{T} \quad (32)$$

Для известного входного меандра коэффициенты A_j вычисляются аналитически, а коэффициенты выходного сигнала B_j вычисляются из экспериментальных данных.

После этого определяется фазово-частотная характеристика (ФЧХ) мембраны:

$$\varphi_j = \arg\left(\frac{B_j}{A_j}\right) \quad (33)$$

Откладывая фазу φ_j в координатах $(j\omega_0, \varphi_j)$, получаем график ФЧХ $\varphi(\omega)$. В расчет принимается только $N=15 \div 25$ первых гармоник, т.к. на высоких частотах амплитуда A_j становится сопоставимой с шумом, а фаза - неустойчивой. В работе [75] показано, что ФЧХ определяется механизмами проникновения и переноса водорода и слабо зависит от случайных факторов. Авторами МКИ был создан банк параметрических моделей $\varphi_p(\omega)$, где p - обобщенный набор (от 1 до 4) модельных параметров. Он позволяет найти параметры p^* оптимальной модели по МНК:

$$p^* = \arg \min \left(\sum_{j=1}^N \left(\varphi_p(\omega_0 j) - \varphi_j \right)^2 \right) \quad (34)$$

С точки зрения АСКУ реализация МКИ - серьезная задача, требующая высокой точности регистрации и управления по времени, ведь ФЧХ чувствительна к сдвигу фазы (т.е. времени) и нарушению периодичности меандра. При типичном периоде меандра $T \approx 100 \text{ с} = 10^5 \text{ мс}$ и относительном временном разрешении $\Delta T/T$ не хуже 12-бит (как у АЦП), получим требование к точности генерации меандра $\Delta T \leq \frac{10^5}{4096} \approx 25 \text{ мс}$. Заметим, что меандр генерируется программно, по часам компьютера. Поэтому время реакции АСКУ должно быть не хуже 10 мс.

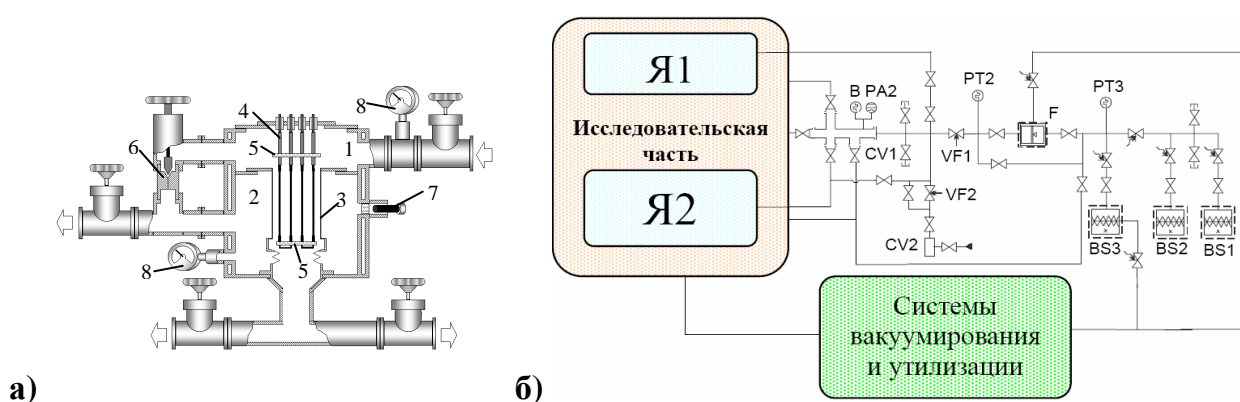
Ячейка Я2 для исследования сверхпроницаемости ИВ

При определенных условиях металлическая мембрана становится сверхпроницаемой, то есть пропускает надтепловые (≥ 1 эВ) атомы водорода, как отверстие той же площади, при этом проницаемость мало зависит от температуры и толщины мембраны [40,41,42,43,44,71,72,73]. Поскольку мембраны остаются непроницаемыми для теплового молекулярного водорода, гелия и любых других газов, они автоматически сжимают проникающий водород и очищают его от любых примесей, включая радиогенный гелий. Явление сверхпроницаемости

обусловлено потенциальным барьером для реэмиссии абсорбированного водорода, создаваемым монослойной неметаллической плёнкой на поверхности металла.

Сверхпроницаемость наблюдалась на мембранах из Fe, Ni, Pd, Nb по отношению к частицам протия и дейтерия [71,72,73]. Исследования с тритием были затруднены в связи с его радиоактивностью. Стенд ПРОМЕТЕЙ впервые дал возможность проводить исследования сверхпроницаемости с тритием.

Для исследования сверхпроницаемости ИВ сквозь металлы используется разработанная в СПбГУТ, в лаборатории проф. А.И. Лившица [40], измерительная ячейка Я2 (Рис.44,а). Она включает в себя входной (1) и выходной объемы (2), разделенные вертикальной цилиндрической мембраной (3) размером $\varnothing 100 \times 180$ мм.



**Рис.44. а) Ячейка Я2 для исследования сверхпроницаемости ИВ.
б) Газовая схема системы напуска и контроля исходной смеси.**

Насос NM3 непрерывно прокачивает поток ИВ через внутренний объем цилиндра (входную часть). Внутри мембраны размещается атомизатор (4) в виде набора из 6 танталовых пластин вдоль мембраны. Фланец атомизатора имеет 6 токовых вводов, рассчитанных на ток до 70 А в стационарном рабочем режиме и до 100 А при кратковременном технологическом нагреве (мощность питания до 1 кВт и до 2 кВт соответственно). Температура атомизатора может достигать до 2300 К, поэтому корпус ячейки и NM2 имеют водяное охлаждение и датчики воды ДГ2, ДГ3. Торцы мембраны прикрыты кварцевыми дисками (5) для экранирования конструктивных элементов ячейки от влияния атомарных частиц. Температуру мембраны измеряют два пирометрических датчика (7) через специальные окна. Насос NM2 непрерывно вакуумирует выходной объем ячейки. Давление измеряют датчики Balzers (8), оно меняется в пределах от $2 \cdot 10^{-7}$ до 10 Па на входной стороне и от $5 \cdot 10^{-7}$ до 10^{-5} Па на выходной. Натекатель VF (6), замыкающий входную и выходную часть ячейки, служит для демонстрации явления рециркуляции ИВ.

Система напуска и контроля исходной смеси

Система напуска и контроля исходной смеси относится к технологической части газовой системы (Рис.44,б) и включает металлогидридные генераторы протия BS1, дейтерия BS2 и трития BS3 с максимальным давлением 104 Па и объемом 50 л (H₂, D₂) и 10 л (T₂). При проведении исследований источники нагревают до температуры 650÷700 К, при этом ИВ поступают в газовые линии. Каждый источник имеет управляемый нагреватель и термопару для контроля температуры. Фильтр F служит для очистки трития от радиогенного гелия, емкость CV1 - для накопления газа в нужном для опытов на ячейках Я1, Я2 количестве. Емкость CV2 требуется для подачи примесей (S, CO, O₂ и др.) в ячейки Я1 или Я2 в процессе экспериментов. Максимальное абсолютное давление в емкостях CV1 и CV2 составляет 1 и 10⁵ Па соответственно. Натекатели VF1, VF2 регулируют напуск ИВ и примесей в ячейку Я2, при расходе газа в пределах от 0 до 10⁻² Па·м³/с. Давления в различных точках регистрируют баротрон В и датчики вакуума РА2, РТ2, РТ3.

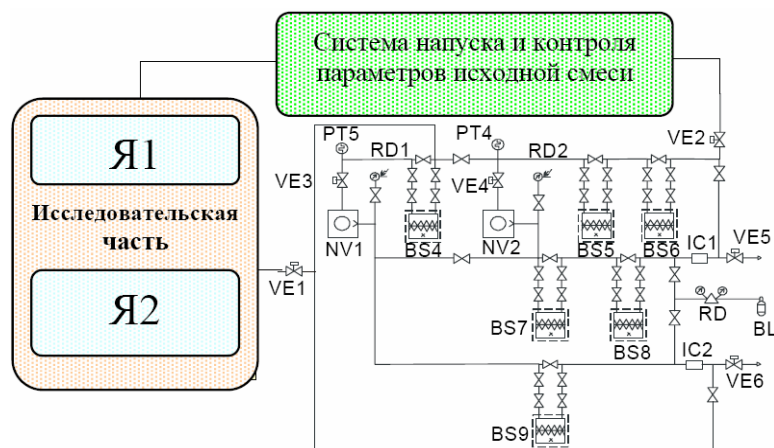


Рис.45. Газовая схема системы утилизации и вакуумирования.

Система вакуумирования и утилизации

Система вакуумирования и утилизации (Рис.45) включает в себя поглотители BS4...BS9, форвакуумные насосы NV1 и NV2, электромагнитные клапаны VE1, VE2, VE3, VE4, VE5, VE6, ионизационные камеры IC1, IC2, редуктор давления RD, баллон газа BL, датчики давления PT4, PT5, RD1, RD2. Подробное описание системы вакуумирования и утилизации и алгоритмов ее работы есть в работе [17].

Автоматизированная система контроля и управления стенда

Большое число измеряемых параметров и управляемых узлов, вместе с потенциальной радиационной опасностью при работе с тритием и потребностью в

удаленном управлении при опасном повышении радиационного фона потребовали создания высоконадежной распределенной сетевой АСКУ. В ее функции входят регистрация и архивирование параметров состояния стенда, управление его исполнительными устройствами, радиометрический контроль по тритию, автоматическая блокировка устройств (для предотвращения аварийных ситуаций) и оповещение персонала (при возникновении радиационной опасности).

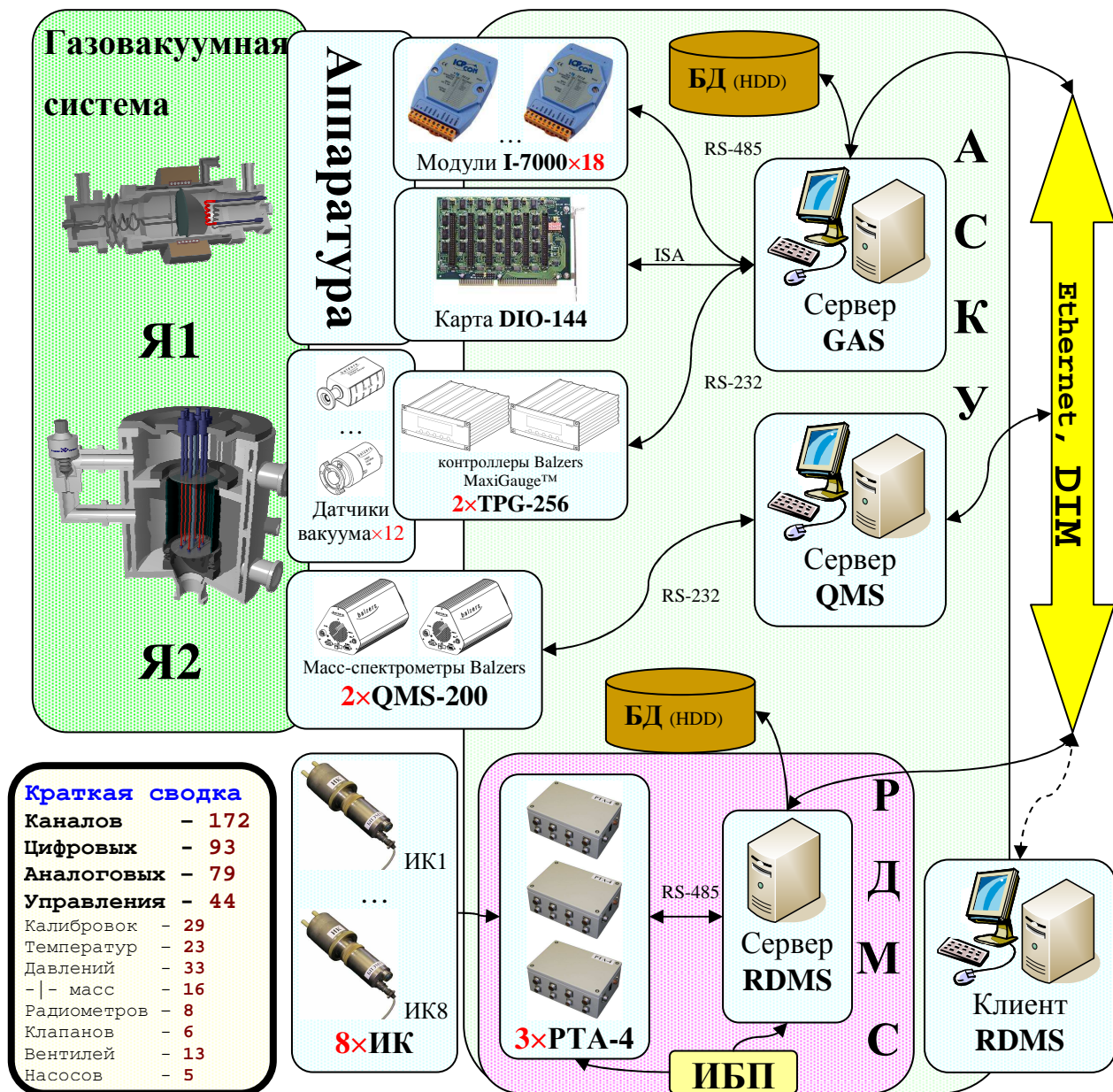


Рис.46. Общая схема АСКУ стенда ПРОМЕТЕЙ.

АСКУ (Рис.46) включает четыре управляющих компьютера (GAS, QMS, RDMS-сервер и клиент), интеллектуальные контроллеры, модули удаленного сбора данных, измерительные карты и специальную аппаратуру (датчики вакуума, радиометры трития), связанные по интерфейсам Ethernet, RS-232, RS-485, ISA. Для

контроля радиационной безопасности создана радиометрическая система RDMS, которая может работать как в автономном режиме, так и в составе АСКУ.

Сервер GAS - главный компьютер стенда ПРОМЕТЕЙ. Он управляет системами напуска и контроля состава исходной смеси, вакуумирования и утилизации, подготовкой и проведением экспериментов на ячейках Я1 и Я2. Он обеспечивает сохранение результатов всех измерений в файловой базе данных и визуализацию всех регистрируемых технологических и физических параметров газовакуумной системы. Сервер осуществляет сбор данных и управление через 18 модулей серии I-7000 ICP DAS, карту цифрового ввода-вывода DIO-144 Advantech, 2 контроллера TPG-256 Balzers и специально разработанную аппаратуру.

Сервер QMS обеспечивает сбор масс-спектрометрических данных, анализ газовой смеси с использованием двух масс-спектрометров QMS-200 Balzers, отключает масс-спектрометры при нарушении вакуума, что необходимо для их безопасной работы, визуализирует масс-спектрометрические данные и передает их серверу GAS. Сервер RDMS обеспечивает управление радиометрической системой, измеряющей объемную активность трития в рабочем помещении и воздуховодах, а также передачу данных на сервер GAS для их визуализации, архивирования и использования в системе блокировок. Клиент RDMS служит для удаленного наблюдения радиационной обстановки из комнаты службы дозиметрического контроля, а также для удаленной настройки радиометров.

Данные с серверов QMS и RDMS передаются на главный компьютер GAS в реальном времени по сети Ethernet, по протоколу DIM [70]. Всё программное обеспечение АСКУ реализовано средствами пакета CRW-DAQ (раздел 1.4).

3.2. Аппаратура для автоматизации стенда ПРОМЕТЕЙ

АСКУ стенда ПРОМЕТЕЙ включает разнообразную аппаратуру для сбора данных и управления: 14 электронных регуляторов мощности для нагревателей источников BS1...BS3, поглотителей BS4...BS9, контейнеров К, К1, К2, фильтра F, ячейки Я1; 14 термопар хромель/алюмель и хромель/копель для измерения температуры источников, поглотителей, фильтра и Я1, по которым контролируется нагрев; термопару для измерения температуры исследуемой мембраны ячейки Я1; усилитель мощности постоянного тока с выходным током до 3 А и напряжением до 10 В для формирования меандра на атомизаторе исследовательской ячейки Я1; три

мощных (до 2.5 кВт) регулируемых источника постоянного тока для атомизатора исследовательской ячейки Я2 с выходным напряжением до 50 В и током до 50 А; два пирометрических датчика температуры для исследовательской ячейки Я2; 3 датчика вакуума на лампах типа ПМТ-4; 8 датчиков для измерения вакуума фирмы Balzers (Pfeiffer); 13 вентилях, снабженных концевиками для контроля их состояния; 6 управляемых электромагнитных клапанов VE1..VE6 и блок управления к ним; 3 магниторазрядных и 2 форвакуумных насосов; два квадрупольных масс-спектрометра QMS-200 фирмы Balzers (Pfeiffer) для анализа газовой смеси, каждый из которых измеряет до 8 массовых концентраций (как функций времени) одновременно; 8 блоков детектирования на основе проточных и диффузионных ионизационных камер объемом 1, 10 и 20 л.

АСКУ стенда ПРОМЕТЕЙ (Таблица 5) содержит 172 измерительных каналов, включая 79 аналоговых каналов (75 - ввода, 4 - вывода), 93 цифровых каналов (53 - ввода, 40 - вывода), 44 канала служат для управления установкой. Температура измеряется в 23 каналах, включая 15 термопар хромель/алюмель и хромель/копель, 3 датчика температуры холодного спая, 3 датчика температуры вакуумных ламп ПМТ и 2 пирометра. Давление измеряется в 33 каналах, включая 12 каналов вакуумных датчиков Balzers (Pfeiffer), 3 вакуумные лампы ПМТ, 2 тензодатчика давления и 16 парциальных давлений различных молекул, измеряемых двумя масс-спектрометрами QMS-200.

Для управления газовой системой применяются 14 управляемых нагревателей, 6 управляемых электромагнитных клапанов, 13 контролируемых вентилях, 5 управляемых насосов (3 магниторазрядных и 2 форвакуумных). Для контроля радиационной обстановки используется 8 радиометрических каналов на ионизационных камерах. Проведение опытов обеспечивают 4 управляемых атомизатора, 3 в ячейке Я2 для исследования сверхпроницаемости и один в ячейке Я1 для исследования проницаемости ИВ. Для поддержки нормального функционирования аппаратуры дополнительно задействован ряд технологических каналов (2 датчика воды для контроля водяного охлаждения насосов, контроль тока для диагностики разрыва спирали и охранная блокировка 14 нагревателей, датчик контакта термопары мембраны Я1 на землю, 3 канала контроля тока ламп ПМТ, 4 канала контроля напряжения и тока атомизаторов).

Таблица 5. Список измерительных каналов АСКУ стенда ПРОМЕТЕЙ.

Измерение и управление	Каналов	Тип канала
ПРОМЕТЕЙ		
Термопарные измерения температуры нагревателей источников	14	аналоговый
Управление нагревателями источников	14	дискретный
Контроль обрыва спиралей нагревателей источников	14	дискретный
Управление релейной блокировкой нагревателей источников	14	дискретный
Измерение температуры холодного спая	3	аналоговый
Управление 3 атомизаторами ячейки Я2	3	аналоговый
Контроль тока 3 атомизаторов ячейки Я2	3	аналоговый
Пирометрические измерения температуры мембраны ячейки Я2	2	аналоговый
Управление нагревателем и атомизатором ячейки Я1	2	аналоговый
Контроль напряжения и тока атомизатора ячейки Я1	2	аналоговый
Термопарные измерения температуры мембраны и ячейки Я1	2	аналоговый
Контроль контакта термопары с мембраной ячейки Я1	1	дискретный
Измерение низких давлений и вакуума датчиками фирмы Balzers	12	RS-232
Контроль давления вакуума лампами ПМТ-4 + контроль их тока	3 + 3	аналоговый
Измерение давлений тензометрическими датчиками	3	аналоговый
Измерение парциальных давлений ИВ масс-спектрометрами QMS-200	16	RS-232
Управление насосами форвакуумными + магниторазрядными	2 + 3	дискретный
Контроль включения насосов форвакуумных	2	дискретный
Контроль тока насосов магниторазрядных	3	аналоговый
Контроль мановакууметров контактных	2	дискретный
Контроль состояний датчиков давления воды	3	дискретный
Контроль состояний ручных вентилях с датчиками положения	26	дискретный
Управление электромагнитными клапанами	6	дискретный
Контроль состояний электромагнитных клапанов	6	дискретный
Контроль объемной активности трития	8	RS-485

При коммутации и работе сильноточных устройств на аппаратуру АСКУ воздействуют мощные электромагнитные помехи. Например, при включении трех атомизаторов Я1 коммутируется ток до 150 А, при напряжении до 50 В и мощности до 7,5 кВт. С учетом 14 нагревателей мощностью до 2 кВт и вакуумных насосов, общая коммутируемая мощность может достигать 20÷30 кВт. Поэтому при разработке АСКУ и выборе аппаратуры много внимания уделялось электрической защите и помехозащищенности измерительных и управляющих каналов,

гальванической изоляции входных и выходных цепей. Приняты также меры для повышения надежности и живучести АСКУ: иерархическая распределенная структура управления, дублирование критически важных узлов (см. раздел 1.4.2.6).

В системе управления используется 18 модулей серии I-7000 фирмы ICP DAS, причины выбора и достоинства которых описаны в разделе 1.4.2.7. Шесть 8-канальных модулей АЦП типа I-7018 опрашивают датчики температур, давлений и т.д. с частотой до 10 Гц. Расположение модулей (на стенке вытяжного шкафа, рядом с источником сигнала) позволяет уменьшить длину сигнальных кабелей и снизить уровень помех. Цифровые модули (четыре I-7043, два I-7053, два I-7060 и I-7050) контролируют состояние и управляют нагревателями, клапанами, насосами. Модуль ЦАП типа I-7024 управляет током накала атомизаторов исследовательских ячеек Я1 и Я2. Контроллер I-7188 и преобразователь интерфейса RS-232/485 типа I-7520 обеспечивают связь компьютера с измерительными модулями.

30 цифровых каналов ввода реализовано на карте DIO-144 фирмы Advantech с интерфейсом ISA. По этим каналам контролируются состояния 13 вентилях (по два канала на вентиль), двух датчиков воды и превышения давления. Они не являются критически важными, не связаны с возможным источником помех (используются датчики типа “сухой контакт”) и поэтому не требуют высокой степени защиты и гальванической развязки. Драйвер DIO-144 был реализован средствами пакета CRW-DAQ на языке DAQ Pascal (раздел 1.4.2.5).

Для вакуумных измерений используется два интеллектуальных контроллера TPG-256 фирмы Balzers. К ним подключается до 12 датчиков вакуума типа IKR-261 или TPR-260. Контроллеры обеспечивают питание датчиков, регистрацию их состояния (подключен, исправен и т.д.), автоматический перевод единиц измерения измеряемого сигнала в давление, отключение датчиков при нарушении вакуума (для предотвращения их разрушения) и другие важные функции. Связь контроллеров с ПК осуществляется через интерфейс RS-232. Драйвер контроллера TPG-256 реализован на языке DAQ Pascal (раздел 1.4.2.5) на основании открытого протокола связи, опубликованного фирмой Balzers, включающего около 30 команд.

Для анализа состава газа на выходе измерительных ячеек используется два квадрупольных масс-спектрометра QMS-200, их обслуживает сервер QMS (Рис.46).

Спектрометр QMS-200 – сложное устройство на основе интеллектуального контроллера, подключаемого к ПК через интерфейс RS-232. Контроллер выполняет аппаратные функции низкого уровня (установка массы, измерение тока электрометра, управление накалом и высоким напряжением и т.д.).

Часть аппаратуры разработана специально для установки ПРОМЕТЕЙ, поскольку подходящего промышленного оборудования не нашлось, либо его применение было экономически неоправданным.

Для управления 14 нагревателями фильтров, источников и ловушек служат 4 блока 4-канальных полупроводниковых регуляторов мощности. Каждый канал оптически изолирован по входу управления, схема на семисторах обеспечивает работу с нагрузкой до 2 кВт. Система управления нагревателями описана ниже.

Для контроля вакуума в газовой системе служат три термоманометрические лампы ПМТ-4, измеряющие температуру подогреваемой термопары, помещенной в вакууме. Давление влияет на температуру термопары из-за условий теплопередачи. Для питания ламп разработан специальный блок, содержащий три регулируемых высокостабильных источника тока, каналы измерения тока и сигнала термопары.

Калибровочная кривая вакуумных ламп существенно нелинейна, особенно при малых давлениях. Зависимость давления P , мбар от термоЭДС u , мВ аппроксимируется полиномом 7 степени в полулогарифмической шкале:

$$Lg(P)_{(u)} = \sum_{k=0}^7 C_k \cdot (u-10)^k \quad (35)$$

Полином вычисляется по МНК (см. раздел 1.4.2.3), при этом относительная точность калибровки составляет $\frac{\sigma(P)}{P} \leq 2\%$.

Для управления 6 электромагнитными клапанами газовакуумной системы и контроля их состояния используется специальный электронный блок, обеспечивающий необходимые токи коммутации. Управление и контроль осуществляются через модули дискретного ввода/вывода серии I-7000.

Три атомизатора ячейки Я2 (мощностью до 2,5 кВт каждый) питаются от трех управляемых источников постоянного тока до 50 А при напряжении до 50 В, защищенных от короткого замыкания, перегрузки по току и перегрева. Управление выполняют модули ЦАП типа I-7024, напряжения и токи на выходе измеряют модули АЦП типа I-7018 (с использованием делителей напряжения и шунтов).

Температуру атомизатора ячейки Я1 регулирует специальный усилитель мощностью до 30 Вт, дающий стабильный управляемый постоянный ток до 3 А при напряжении до 10 В. Усилитель управляется через модуль ЦАП типа I-7024, напряжение и ток на выходе усилителя измеряются модулями АЦП типа I-7018. Усилитель мощности генерирует меандр в методе МКИ.

3.3. Программное обеспечение для автоматизации экспериментов на стенде ПРОМЕТЕЙ

Программное обеспечение для АСКУ стенда ПРОМЕТЕЙ разработано на базе описанного в разделе 1.4 пакета CRW-DAQ [19], который устанавливается и запускается на всех машинах АСКУ (Рис.46). На трех серверах (GAS, QMS, RDMS) пакет и измерительная программа запускаются автоматически, при старте ОС. На клиентских ПК пакет запускается по желанию пользователя.

Прикладная часть ПО для АСКУ на каждой машине своя и содержит всю специфику управляемой системы. Прикладной код включает в себя (суммарно, на 4-х компьютерах) около 50 файлов конфигурации, 35 файлов калибровок измерительных каналов, 13 файлов описаний мнемосхем, 360 файлов изображений для них, 50 файлов с программами на языке DAQ Pascal.

Ряд устройств в составе АСКУ (радиометры РТА-4, регуляторы мощности, контроллеры TPG-256 и QMS-200) потребовали создания специальных драйверов, т.к. фирменные драйверы отсутствовали или не удовлетворяли требованиям данной задачи. Все драйверы и прикладные управляющие программы реализованы встроенными средствами пакета CRW-DAQ, в основном на языке DAQ Pascal (раздел 1.4.2.5).

3.3.1. Управление газовакуумной системой

АСКУ стенда управляет всеми необходимыми параметрами систем напуска, вакуумирования и утилизации: регистрирует давления и температуры, контролирует состояние вентилях и клапанов, управляет нагревателями источников и поглотителей ИВ, насосами и клапанами, автоматически блокирует выход газов в атмосферу при повышении радиационного фона.

Наиболее сложной частью АСКУ является газовакуумная подсистема для контроля состояния вентилях, измерения температур и давлений, управления

клапанами, насосами, масс-спектрометрами, ячейками Я1 и Я2. Управление нагревателями, атомизаторами, масс-спектрометрами выделено в отдельные подсистемы в её составе. Газовакуумной подсистемой управляет сервер GAS, обменивающийся данными с масс-спектрометрическим сервером QMS и радиометрическим сервером RDMS по протоколу DIM.

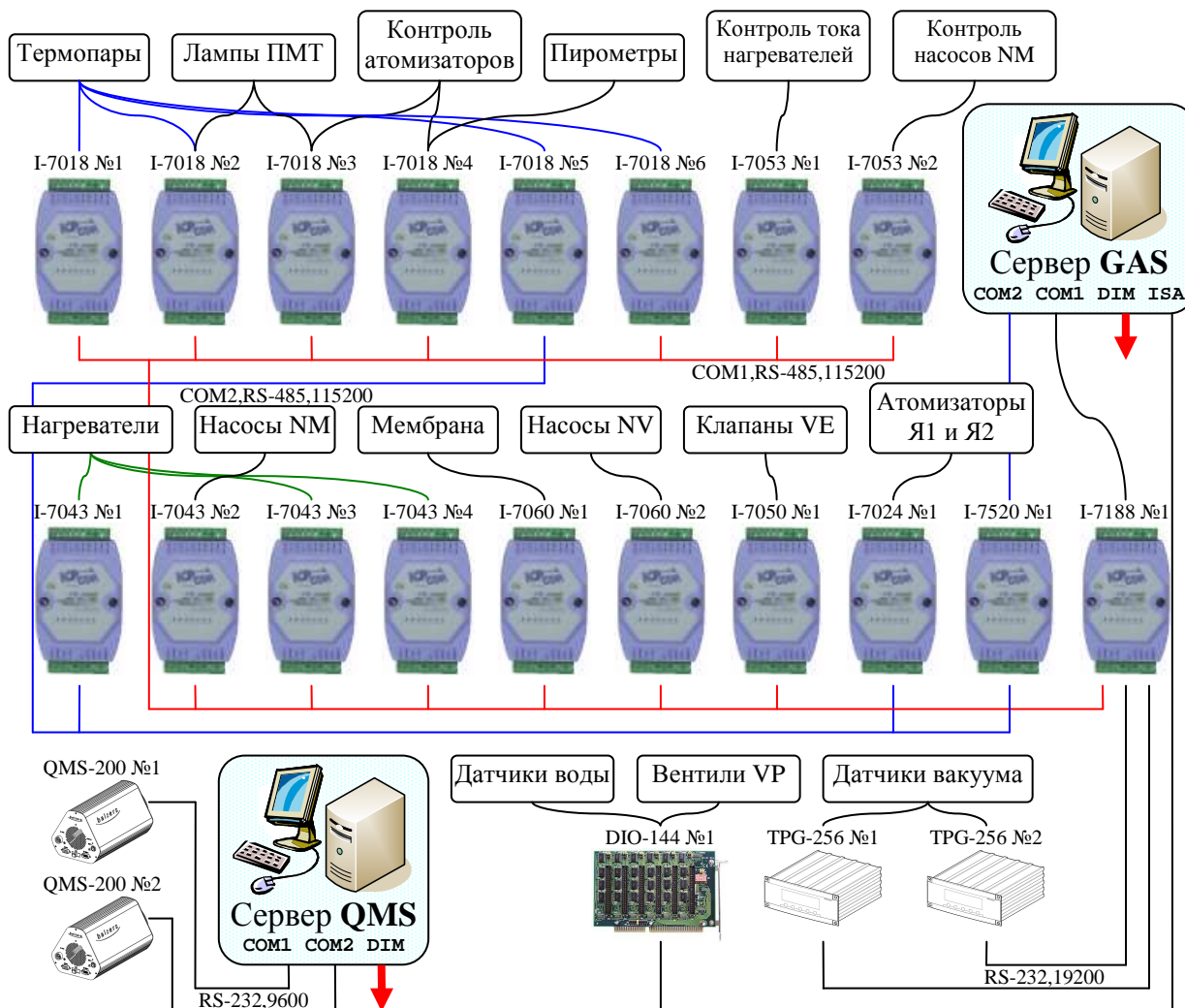


Рис.47. Сетевая схема газовакуумной и масс-спектрометрической системы.

Газовакуумная и масс-спектрометрическая части АСКУ имеют сложную сетевую структуру (Рис.47). 13 измерительных модулей I-7000, с периодом опроса 100÷500 мс на канал, и два контроллера TPG-256 подключены через контроллер I-7188 к порту COM1 сервера GAS. Три модуля I-7000, с периодом опроса 10÷100 мс на канал, подключены к порту COM2 через преобразователь интерфейса RS-232/485 типа I-7520. Разделение модулей по портам обусловлено требованиями по скорости опроса модулей. Модули управления нагревателями и атомизаторами требуют повышенной частоты опроса для генерации управляющего меандра ШИМ. Скорости опроса других каналов определяются возможностями измерительных

устройств и составляют: 9600 бит/сек для масс-спектрометров, 19200 бит/сек для контроллеров вакуума и 115200 бит/сек для модулей I-7000. ПО для управления газовакуумной системой представляет собой набор прикладных программ на языке DAQ Pascal. Каждая из них работает в своем потоке, выполняя простую, четко поставленную задачу, что позволяет упростить программы и сделать их более надежными. Параллельная работа нескольких простых программ позволяет реализовать более сложные алгоритмы управления. Например, задачу управления нагревателями совместно решают 5 потоков (драйвер COM-порта Uart, драйвер модулей I-7000 Adams, модуль калибровок &Pro_Temp, стабилизатор &UH_Stab, генератор ШИМ меандра &UH_Pulse).

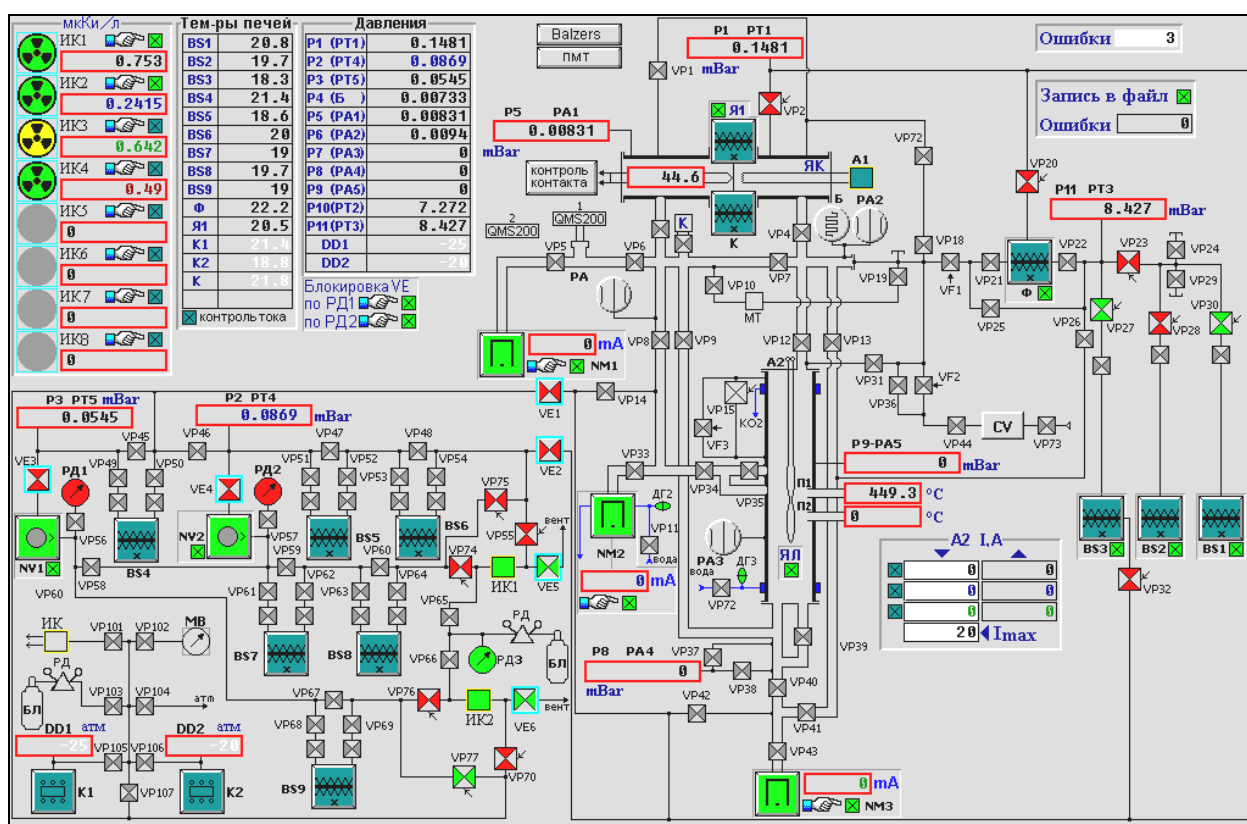


Рис.48. Главная мнемосхема газовакуумной системы стенда ПРОМЕТЕЙ на сервере GAS.

Графический интерфейс АСКУ выполнен в виде активных мнемосхем. Главная мнемосхема газовакуумной системы на сервере GAS близко соответствует привычной для исследователей газовой схеме стенда и содержит активные элементы для отображения состояния узлов стенда и управления ими (Рис.48). Через неё можно также вызвать другие управляющие окна, например, при щелчке на изображении нагревателя вызывается мнемосхема системы управления нагревателями (Рис.51).

Измеряемые параметры отображаются в текстовой, мнемонической и графической форме в режиме реального времени. Щелчком мыши на числовом поле, отображающем объемную активность, температуру или давление датчика, можно вызвать окно с графиком данной величины, а щелчком на изображении электромагнитного клапана или насоса можно включить или выключить его.

3.3.2. Управление нагревателями фильтров, источников и ловушек ИВ

АСКУ стенда имеет 14 каналов автоматизированного управления нагревом источников ИВ, поглотителей и фильтров (Рис.49,а). Она управляет нагревателями, стабилизирует их температуру, контролирует обрыв цепи, автоматически блокирует нагрев при повышении температуры, давления ИВ и радиационного фона. Температуру нагрева измеряют термопары хромель-копель или хромель-алюмель, их термоЭДС регистрируется модулем АЦП I-7018.

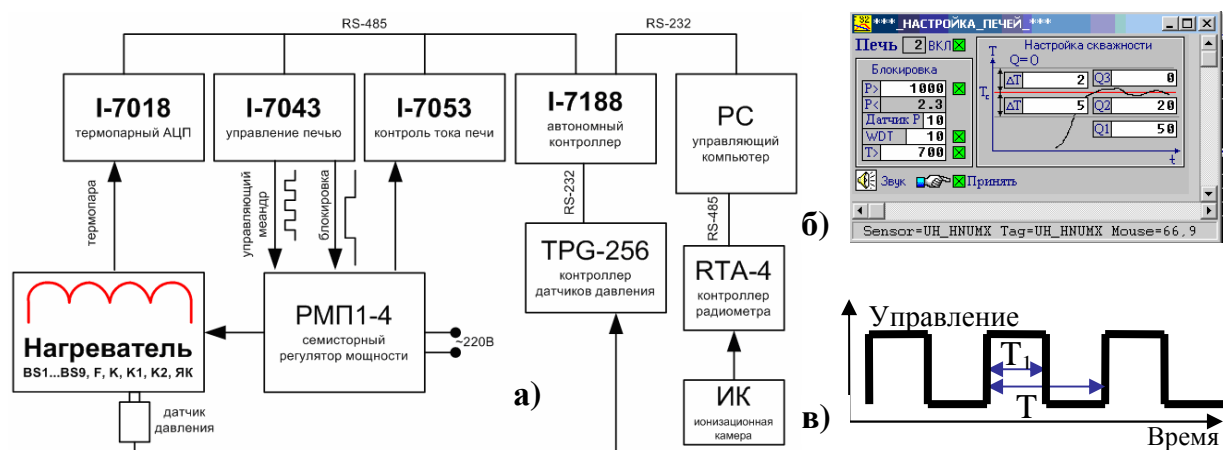


Рис.49. а) Блок-схема одного канала управления нагревом; б) мнемосхема для задания параметров канала управления нагревом; в) вид управляющего сигнала ШИМ.

Нагревом управляют полупроводниковые регуляторы РМП1-4 с максимальной выходной мощностью до 2 кВт, работающие в ключевом режиме, что обеспечивает низкие потери энергии и снимает проблему охлаждения электронных блоков. Регулирование осуществляется методом широтно-импульсной модуляции (ШИМ). Управляющий меандр ШИМ с постоянным периодом и переменной скважностью формируется компьютером программно (Рис.49,б) и подается на вход регулятора через модуль дискретного вывода I-7043. Такая схема формирования ШИМ резко удешевляет канал нагрева, но повышает требования к ПО для АСКУ.

Мощность, выделяемая нагревателем, определяется отношением в процентах длительности состояния «включен» к длительности периода модуляции (Рис.49,в)

$$Q = \frac{T_1}{T} \cdot 100\% \quad (36)$$

Объекты регулирования массивны и имеют типичное время нагрева ~10 минут, поэтому меандр ШИМ эффективно сглаживается. При периоде ШИМ 2÷5 сек температура стабилизируется с точностью $\pm 1 \div 2^\circ\text{C}$ (при температурах ~500°C).

Датчик тока нагрузки в составе канала регулирования диагностирует обрыв цепи или пробой полупроводникового регулятора мощности. Он подключается к компьютеру через многоканальный модуль дискретного ввода I-7053. В системе блокировок канала регулирования нагрева используются также датчики давления ТРГ-256 и радиометрические каналы РТА-4.

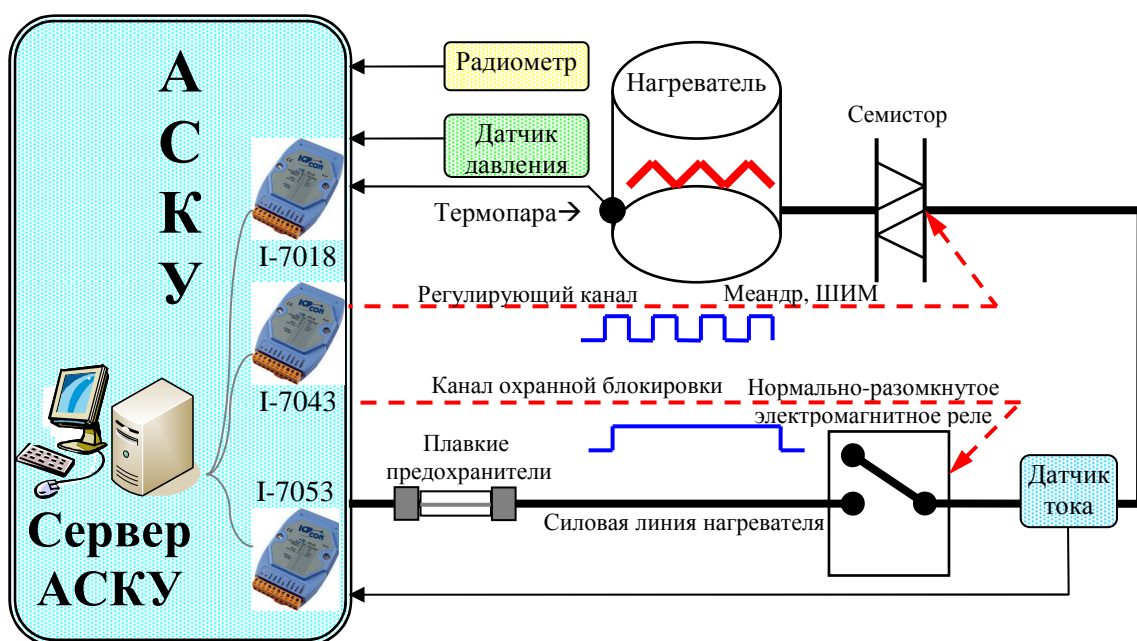


Рис.50. Устройство защиты канала регулирования температуры.

Особое внимание при разработке канала нагрева уделялось надежности и отказоустойчивости системы управления (Рис.50). Она должна не допускать перегрева объектов, а также блокировать нагреватель по превышению давления и уровня радиации. Для повышения надежности управляющие элементы (семисторы, реле) имеют значительный запас по мощности и току. Для отказоустойчивости в силовую цепь включены два управляющих элемента – полупроводниковый регулятор мощности и нормально-разомкнутое электромагнитное реле, которое выполняет охранную роль и обычно постоянно включено. За счет избыточности (дублирующего канала релейной блокировки) система не теряет управляемость даже при коротком замыкании семистора в результате пробоя, при этом реле используется либо для аварийного отключения нагревателя, либо как резервный

канал регулирования, если отключение нагрева в данный момент нежелательно. Нормально–разомкнутое реле гарантирует также отключение силовой линии нагревателя при отсутствии питания в системе управления нагревателями.

Источник ИВ предохраняют от перегрева: при вводе пользователем явно ошибочной целевой температуры – блокировка по температуре; при потере связи или отказе модулей АЦП I-7018 – блокировка по программному сторожу; при поломке термопары – блокировка по датчику давления; при обнаружении утечки радиации – блокировка по каналу радиометра. При блокировке нагрев отключается, а выход трития из источника прекращается. Канал нагрева защищен также от сбоя управляющего компьютера, т.к. в модулях серии I-7043 есть сторожевой таймер, который при потере связи с ПК автоматически отключает регулятор мощности и включает релейную блокировку. Таким образом, система имеет высокую живучесть: отказ любого отдельного её элемента не ведет к аварии.

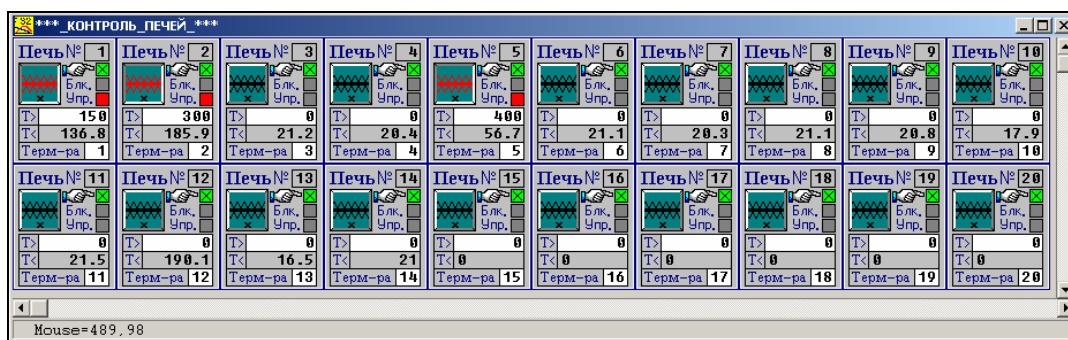


Рис.51. Мнемосхема управления нагревателями установки ПРОМЕТЕЙ.

Логика работы нагревателей примерно одинакова, поэтому управление ими выделено в отдельную подсистему с мнемосхемой (Рис.51), позволяющей включать и отключать нагреватели, задавать желаемую температуру стабилизации и номер контрольной термопары, а также вызывать окно с графиками температур и диалог настройки параметров канала (Рис.49,б).

3.3.3. Измерения на исследовательских ячейках

Исследования на ячейке Я2 требуют измерения давления на входной и выходной стороне ячейки, температуры 3 атомизаторов, 2 пирометров, управления током 3 мощных атомизаторов, насосами и клапанами газовакуумной системы. Управление Я2 осуществляется через главную мнемосхему (Рис.48).

Опыты на ячейке Я1 требуют измерения давления на входной и выходной стороне ячейки, измерения и стабилизации температуры мембраны, управления

током атомизатора и генерации высокоточного по времени меандра, управления насосами и клапанами газовакуумной системы. Часть функций управления Я1 реализуются через общую мнемосхему (Рис.48). Управление атомизатором для методики МКИ выделено в отдельную мнемосхему (Рис.52).

Меандр, т.е. напряжение $V(t)$ атомизатора, имеет вид

$$V(t) = \begin{cases} B, & t \in [n \cdot T, (n+0.5) \cdot T] \\ B + A, & t \in [(n+0.5) \cdot T, (n+1) \cdot T] \end{cases}, n = 0, 1, \dots, N \quad (37)$$

где B - “подложка”, A - амплитуда, n - номер цикла МКИ. Такой меандр улучшает качество входного концентрационного импульса по сравнению с меандром, меняющимся от 0 до максимума. Накал атомизатора имеет значительную инерцию (1-2 сек) при нагреве от комнатной температуры. В то же время степень диссоциации атомизатора резко меняется от нуля до максимума в сравнительно узком диапазоне температур. Если держать его в подогретом состоянии чуть ниже порога диссоциации, а затем нагревать до температуры выше порога диссоциации, то инерционность атомизатора падает в несколько раз.

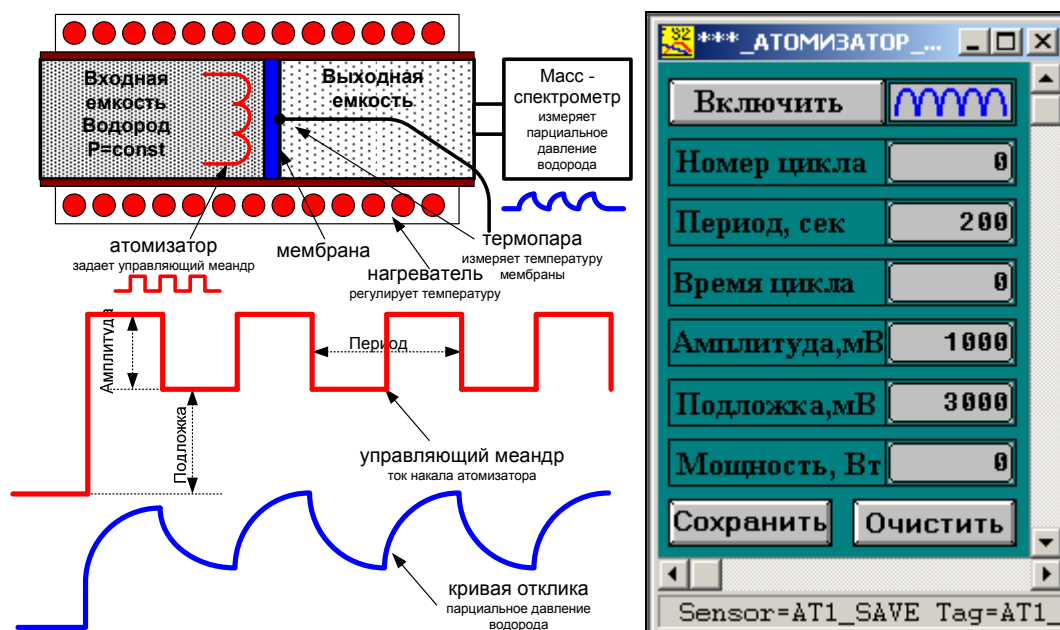


Рис.52. Меандр атомизатора ячейки Я1 и мнемосхема для управления им.

Как и в других подсистемах АСКУ, в управлении ячейкой Я2 принимает участие много взаимодействующих программных потоков, имеющих разный приоритет и период опроса. При этом сложная задача решается параллельной работой нескольких простых потоков.

3.3.4. Масс-спектрометрические измерения

Проницаемость мембран для ИВ в ячейках Я1, Я2 измеряется двумя квадрупольными масс-спектрометрами QMS-200. При постоянной скорости откачки она пропорциональна парциальному давлению ИВ, поэтому масс-спектрометры измеряют интенсивность нескольких интересующих изотопных масс, как функций времени, а не сканируют по массе, как при анализе состава. АСКУ станда управляет режимом работы спектрометров, регистрирует масс-спектры и парциальные давления ИВ как функции времени, автоматически отключает масс-спектрометры при опасном для них повышении давления газа в системе. Спектрометры обслуживаются сервером QMS, а измеренные данные передаются серверу GAS по протоколу DIM (Рис.46).

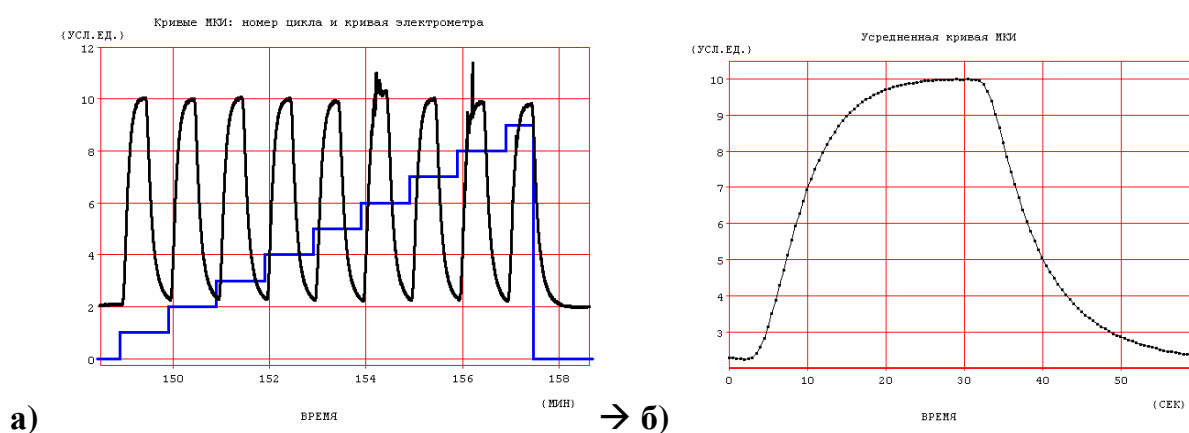


Рис.53. а) Кривая номера цикла и концентрации; б) Усредненная кривая отклика МКИ.

Фирменное ПО спектрометров оказалось непригодным для решаемой задачи. Фирменная программа закрыта и не передает данные в реальном времени, а для реализации МКИ данные масс-спектрометра должны быть четко синхронизованы с атомизатором, генерирующим концентрационные импульсы. В опытах на ячейке Я2 требуется измерять набор интенсивностей $\{I_{m_1}(t), I_{m_2}(t), \dots, I_{m_n}(t)\}$ как функций времени для ряда фиксированных масс изотопов $\{m_1, m_2, \dots, m_n\}$. Аппаратура и система команд спектрометра допускает требуемые измерения, но фирменное ПО рассчитано на сканирование по массам, что здесь не подходит. По этим причинам на основании опубликованного фирмой Balzers протокола связи, включающего около 80 команд, для проведения измерений создан специальный драйвер и программа управления на языке DAQ Pascal. Она позволяет измерять

одновременно до восьми кривых $\{I_{m_1}(t), I_{m_2}(t), \dots, I_{m_8}(t)\}$ интенсивности масс изотопов $\{m_1, m_2, \dots, m_8\}$. Аппаратно QMS-200 позволяет одновременно измерять до 64 масс, однако при этом снижается частота опроса и возрастает шум, поэтому было решено измерять не более 8 масс. На практике число измеряемых масс меняется от 1 (для чистых ИВ) до 6÷8 (для смесей) при периоде опроса 10÷100 мс на канал.

При измерениях (Рис.53,а) регистрируется и сохраняется для анализа номер цикла (меандра) $n(t)$, в виде ступеньки с единичным шагом, и концентрационный отклик $c(t)$, зарегистрированный масс-спектрометром. Кривая номера цикла $n(t)$ дает метки времени $\{t_1, t_2, \dots, t_n\}$ - каждая ступенька соответствует началу периода концентрационного меандра. Полученные метки времени используются для вычисления точного значения T периода меандра, его погрешности $\sigma(T)$ и усредненной кривой МКИ $c_{MKИ}(t)$. Период меандра T и находится как

$$T = \frac{t_n - t_1}{n - 1}, \quad \sigma(T) = \sqrt{\frac{\sum_{i=1}^{n-1} (t_{i+1} - t_i - T)^2}{n - 1}} \quad (38)$$

Усредненная кривая отклика МКИ (Рис.53,б) находится путем сдвига по времени и усреднения периодов измеренного концентрационного отклика $c(t)$:

$$c_{MKИ}(t) = \frac{\sum_{i=1}^{n-1} c(t - t_i)}{n - 1}, \quad t \in [0, T] \quad (39)$$

На практике (Рис.53,а) часть данных бывает повреждена помехами, поэтому предусмотрено исключение из расчетов поврежденных периодов меандра. Первый и последний меандр тоже исключаются из-за влияния переходных процессов.

Для управления масс-спектрометрическими измерениями и расчета кривой усредненного отклика МКИ создан интерфейс на основе активных мнемосхем и диалоговых окон. Центр управления осуществляет общий контроль масс-спектрометра: старт измерений, запись, визуализацию и передачу данных по сети на основной сервер GAS. Перед стартом параметры измерений задаются в диалоге, определяющем режим работы, список измеряемых масс $\{m_1, m_2, \dots, m_n\}$ и частоту опроса. После измерений вызывается формулятор, который по кривым номера цикла $n(t)$ и концентрации $c(t)$ формирует усредненную кривую отклика МКИ, которая записывается и передается в программу обработки Fourier, разработанную И. Габисом и Е.А. Евардом для расчета модельных параметров МКИ [74,75].

3.3.5. Система радиометрического контроля

Система радиометрического контроля (RDMS) - автономная подсистема АСКУ для контроля объемной активности (ОА) трития в воздухе рабочего помещения и в технологических коммуникациях установки. Контроль выполняется с помощью автоматизированных 4-х канальных радиометров трития РТА-4 [58], которые зарегистрированы в Гос. реестре средств измерений под №27238-04, сертификат №18083, и позволяют непрерывно наблюдать ОА трития в 4 точках контроля с частотой опроса до 10 Гц, дают возможность подключения внешнего блока аварийной сигнализации, срабатывающего по заданным уровням. Радиометр работает как автономно, так и в составе АСКУ, с подключением к ПК через интерфейс связи RS-485. RDMS обеспечивает восемь точек контроля (Таблица 6).

Таблица 6. Точки контроля радиометрической системы.

Камера	Тип	Объем, л	Комментарий
ИК1	Поточная	1	Выхлоп насоса NV1
ИК2	Поточная	1	Выхлоп насоса NV2
ИК3	Поточная	1	Воздуховод вытяжного шкафа исследовательской секции
ИК4	Поточная	1	Воздуховод вытяжного шкафа секции утилизации
ИК5	Поточная	1	Общий воздуховод (выход в атмосферу)
ИК6	Диффузионная	10	Рабочее помещение
ИК7	Поточная	1	Воздуховод вытяжного шкафа
ИК8	Диффузионная	20	Блок детектирования А1905-П71

Задачей RDMS является измерение ОА трития в газовых коммуникациях и воздушной среде рабочих помещений, своевременное оповещение персонала о радиационной опасности и блокировка узлов газовакуумной системы при превышении допустимого уровня объемной активности. При работе на стенде контроль ОА и архивирование данных проводится непрерывно, пока в системе есть тритий. Для повышения надежности работы радиометры запитываются отдельной линией, снабженной источником бесперебойного питания (ИБП). Радиометрическую систему обслуживает сервер RDMS - отдельный ПК без монитора и клавиатуры, запитанный от этого же ИБП.

ПО для управления RDMS написано на языке DAQ Pascal (раздел 1.4.2.5). Сервер управляет режимами работы радиометров, сбором данных и их архивированием в файловую базу данных на жестком диске, дает сетевой доступ в реальном времени к радиометрическим данным через технологии DIM и Web [70].

К серверу может подключаться любое число клиентов RDMS или Web. Клиенты RDMS используют технологию DIM, причем у них должен быть установлен пакет CW-DAQ и клиентская конфигурация RDMS. Web-клиенты используют штатный Web-обозреватель и не требуют установки дополнительного ПО. Из-за высоких накладных расходов и довольно низкой скорости Web-технологии применяются только как дополнение к высокоскоростной технологии DIM, которая принята как основной протокол связи в пакете CRW-DAQ.

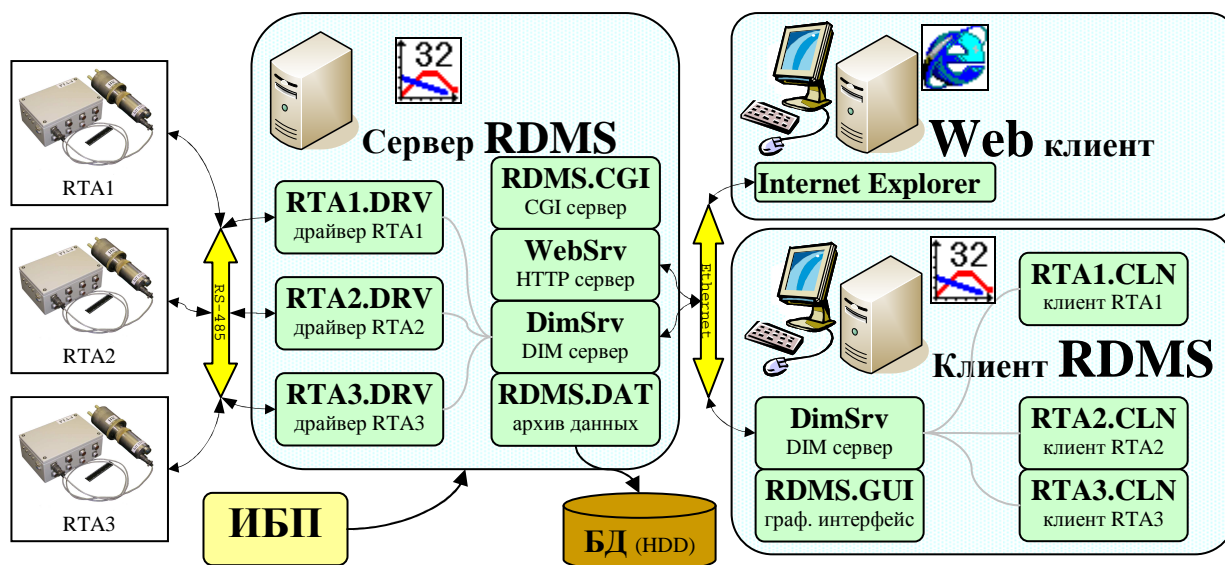
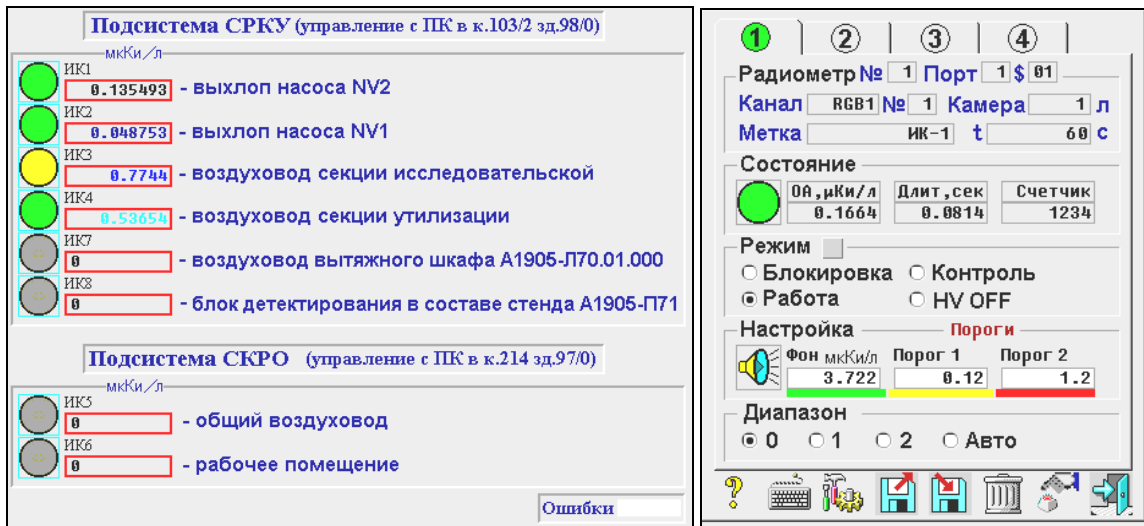
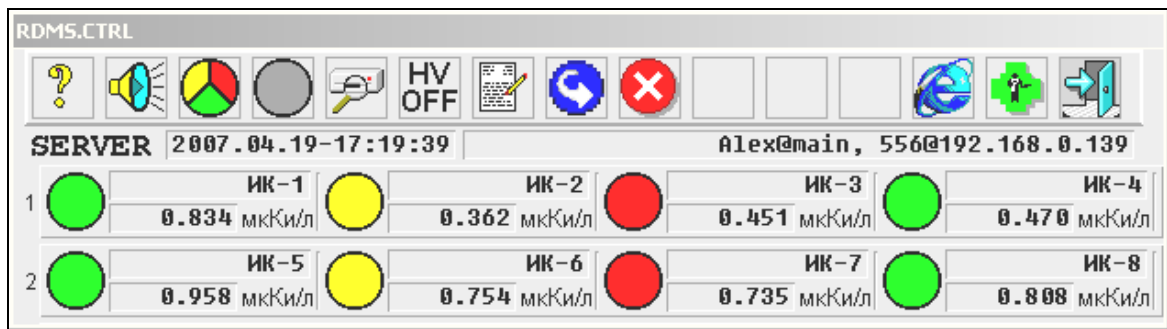


Рис.54. Программная структура радиометрической системы.

В составе сервера RDMS (Рис.54) параллельно работает 7 программных потоков в виде прикладных программ на языке DAQ Pascal. Каждый радиометр обслуживает свой драйверный поток (RTA1.DRV...RTA3.DRV), управляющий сбором данных по сети RS-485. Поток DimSrv дает клиентам сетевой доступ к данным по протоколу DIM. Поток RDMS.DAT архивирует измеряемые данные на диске. Потоки WebSrv и RDMS.CGI совместно обеспечивают работу Web сервера: первый обрабатывает HTTP запросы, второй генерирует динамические HTML страницы и обеспечивает систему прав доступа к серверу. В составе клиента RDMS работает 5 программных потоков, также написанных на DAQ Pascal. Каждый радиометр имеет свой поток (RTA1.CLN... RTA3.CLN), формирующий клиентскую модель удаленного радиометра. Поток DimSrv принимает данные от сервера по протоколу DIM и отправляет команды от пользователя. Поток RDMS.GUI отображает данные и транслирует действия оператора в команды, передаваемые по сети.



а) Центр управления и главная мнемосхема.

б) Свойства канала радиометра.

Рис.55. Основные мнемосхемы радиометрической системы стенда ПРОМЕТЕЙ.

Для RDMS разработан графический интерфейс на активных мнемосхемах. Главная мнемосхема (Рис.55,а) отображает текущее состояние каждого канала в виде цветового индикатора и числового поля объемной активности в единицах измерения мкКи/л. Щелчком мыши на числовом поле можно открыть график кривых объемной активности канала. Серый цвет индикатора обозначает отключенный канал измерения, зеленый цвет означает, что радиационный фон в пределах нормы. Желтый цвет сигнализирует о неопасном, а красный - об опасном превышении радиационного фона. Щелчком мыши на индикаторе радиометра вызывается мнемосхема для настройки его канала задания режимов работы и установки порогов сигнализации (Рис.55,б).

Измеренные данные по ОА трития передаются для удаленного наблюдения за радиационной обстановкой на ПК клиента RDMS в пультовой дозиметрического контроля (Рис.46) и на сервер GAS для работы системы блокировок.

3.4. Результаты экспериментов на стенде ПРОМЕТЕЙ

За 8 лет постоянной эксплуатации автоматизированного стенда ПРОМЕТЕЙ, коллективами под руководством А.А.Юхимчука (ВНИИЭФ), А.А. Курдюмова (НИИФ СПбГУ) и М.Е. Ноткина (СПбГУТ) были получены важные физические результаты.

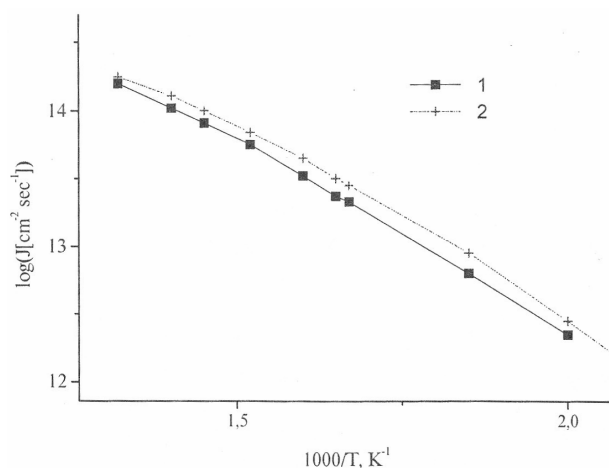


Рис.56. Диффузия протия в никеле [40]: 1-данные ПРОМЕТЕЙ, 2 – данные НИИФ СПбГУ.

Первые работы [40] по исследованию сверхпроницаемости протия через ниобиевую мембрану и диффузии протия в никеле подтвердили работоспособность автоматизированного стенда ПРОМЕТЕЙ и возможность проведения на нем систематических исследований проницаемости и сверхпроницаемости изотопов водорода через конструкционные материалы (например, см. Рис.56).

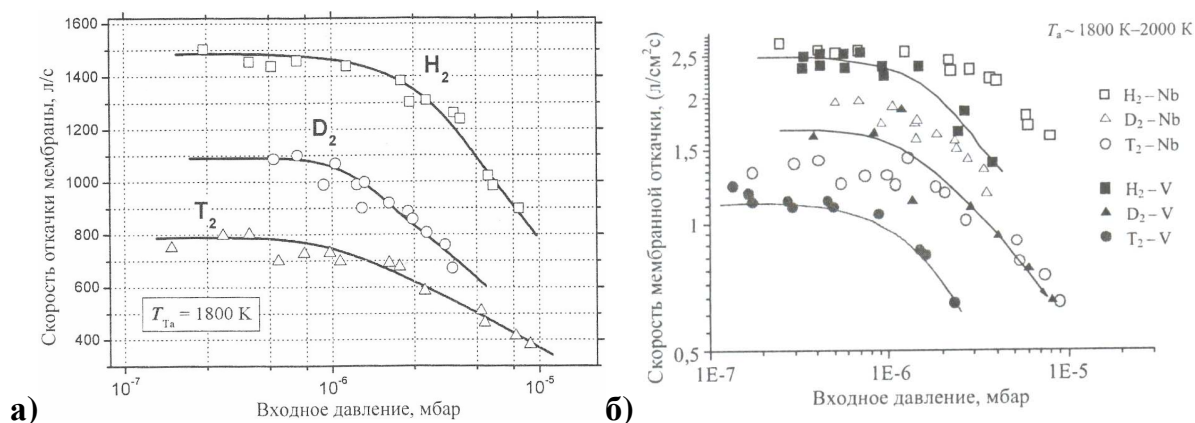


Рис.57. Скорость мембранной откачки ИВ на стенде ПРОМЕТЕЙ при термической атомизации а) на ниобии при $T=1800$ К [44]; б) на ванадии при $T=1800-2000$ К [43].

В экспериментах [41,42] по изучению сверхпроницаемости изотопов водорода через ниобиевые цилиндрические мембраны большой площади впервые сверхпроницаемость была подтверждена для трития. Была измерена скорость мембранной откачки для протия, дейтерия и трития (Рис.57). Показано, что наблюдаемые максимальные скорости мембранной откачки для изотопов водорода

(Рис.57,а) пропорциональны газокинетическому фактору $1/\sqrt{m}$ для молекулярной массы m и составляют 2,5 л/(см²·с) для протия, 1,8 л/(см²·с) для дейтерия и 1,5 л/(см²·с) для трития, что находится в согласии с теорией. Опыты также продемонстрировали возможность эффективной откачки, разделения ИВ от гелия и остаточных газов, компрессии и рекуперации ИВ на мембранах из ниобия.

В опытах [43] по сверхпроницаемости тепловых атомов ИВ, включая тритий, через ванадиевую цилиндрическую мембрану продемонстрирована возможность эффективной откачки, компрессии и рекуперации ИВ и даны оценки удельной скорости откачки (Рис.57,б) и предельной степени компрессии мембранного насоса на основе сверхпроницаемой мембраны из ванадия. Показано, что максимальная удельная скорость мембранной откачки при температуре атомизатора $T=1800\div 2000$ К и давлении $10^{-7}\div 10^{-5}$ мбар для мембраны из ванадия составляет 2,4 л/(см²·с) для протия, 1,7 л/(см²·с) для дейтерия и 1,2 л/(см²·с) для трития.

На стенде исследовали применение эффекта сверхпроницаемости для откачки и рециркуляции топливной D/T смеси в ТЯР с помощью мембран из ванадия, ниобия или тантала [44,45]. Была продемонстрирована возможность успешной откачки, компрессии и рециркуляции трития с помощью сверхпроницаемых ниобиевых и ванадиевых мембран.

Установка ПРОМЕТЕЙ использовалась также для получения образцов, содержащих радиогенный гелий [46], необходимых для исследования физико-механических свойств конструкционных материалов.

3.5. Выводы

Решены вопросы безопасного управления тритиевым комплексом низкого давления исследовательского стенда ПРОМЕТЕЙ и автоматизации экспериментов на исследовательских ячейках по изучению взаимодействия изотопов водорода с конструкционными материалами и сверхпроницаемости изотопов водорода через металлические мембраны.

Распределенная система контроля и управления комплекса обеспечивает подготовку газовых смесей, регистрацию и архивирование технологических и физических параметров, масс-спектрометрические измерения, радиометрический

контроль по тритию, автоматическую блокировку устройств и оповещение персонала при возникновении радиационной опасности.

Длительный (с 2001 по 2010 г.) опыт эксплуатации стенда ПРОМЕТЕЙ при проведении экспериментов по тематике «Взаимодействие изотопов водорода с конструкционными материалами» подтвердил правильность принципов, заложенных в разработку аппаратного и программного обеспечения АСКУ.

С помощью автоматизированного стенда ПРОМЕТЕЙ впервые явление сверхпроницаемости подтверждено для трития, измерены скорости мембранной откачки на мембранах из ванадия и ниобия. Полученные результаты имеют важное научное и практическое значение, например, для топливного цикла ТЯР.

Заключение

Для решения задач автоматизации тритиевых комплексов исследовательских установок ТРИТОН, АКУЛИНА и ПРОМЕТЕЙ создан инструментальный программный пакет CRW-DAQ [47,48,50,19], содержащий встроенные языки прикладного программирования и средства отладки, инструменты для online и offline обработки и сглаживания данных, калибровки каналов, высоконадежные встроенные драйверы для разнообразной измерительной аппаратуры и управления.

Пакет обеспечивает полный цикл разработки, отладки и поддержки АСКУ и позволяет в короткие сроки создавать высоконадежные многопоточные, многопроцессные и многомашинные распределенные системы управления. Его достоинствами являются открытость, простота освоения, ясная структура, высокая скорость создания прикладных программ, их безопасность и отказоустойчивость.

Создано ПО для распределенной автоматизированной системы контроля и управления тритиевым комплексом высокого давления установки ТРИТОН для исследований процессов МК ядерных реакций в смесях ИВ в широком диапазоне температур, плотностей и концентраций ИВ. АСКУ комплекса [8] обеспечивает измерение большого числа аналоговых и дискретных параметров и управление узлами комплекса, измерение изотопного и молекулярного состава смеси ИВ, измерение и стабилизацию параметров мишени в процессе измерений, радиометрический контроль ОА трития в газовых линиях и рабочих помещениях.

Созданный аппаратно-программный комплекс для подготовки газовой смеси и управления мишенью работал надежно. Принятый подход к построению АСКУ позволял быстро адаптироваться к изменениям, возникающим при подготовке к новым экспериментам, обновлении аппаратуры и вводе новых датчиков и каналов.

Под управлением АСКУ на установке ТРИТОН проведен большой цикл измерений параметров МК, получены уточненные и новые данные для параметров цикла МК в дейтерии, в двойной D/T смеси и в тройной H/D/T смеси в широком диапазоне температур, давлений и концентраций [24,25,26,27,28,29,30,31,32,33].

Для экспериментов по получению и изучению нейтронно-избыточных ядер ^4H и ^5H , проводящихся в ЛЯР ОИЯИ, созданы автоматизированные системы для управления напуском изотопов водорода в ионный источник циклотрона У-400М [12,13] и управления газовым комплексом жидкой тритиевой мишени [14,15,16].

Обеспечено тонкое регулирование потоков ИВ, что позволило реализовать точное поддержание требуемых рабочих параметров пучка тритонов в жестких условиях функционирования электронной аппаратуры на ускорителе [12,13]. Обеспечено управление процессом заполнения мишени и утилизации трития, стабилизация температуры мишени в ходе экспериментов, а также радиационный контроль объемной активности трития в газовых коммуникациях комплекса и рабочих помещениях [14,15,16].

С использованием разработанных автоматизированных систем проведены эксперименты по изучению резонансных состояний ядер ^4H и ^5H , образующихся при облучении дейтериевой или тритиевой мишени пучком тритонов, в которых получены новые физические результаты [34,35,36,37,38,39].

Решены вопросы безопасного управления тритиевым комплексом низкого давления исследовательского стенда ПРОМЕТЕЙ и автоматизации экспериментов на исследовательских ячейках для изучения взаимодействия ИВ с КМ и изучения сверхпроницаемости ИВ через металлические мембраны. АСКУ комплекса [18] обеспечивает подготовку газовых смесей, регистрацию и архивирование технологических и физических параметров, масс-спектрометрические измерения, радиометрический контроль по тритию, автоматическую блокировку устройств и аварийное оповещение персонала при возникновении радиационной опасности.

На автоматизированном стенде ПРОМЕТЕЙ проведен большой цикл экспериментов по изучению сверхпроницаемости ИВ через металлические мембраны, проницаемости и накопления ИВ в металлах и КМ [40,41,42,43, 44,45,46]. Длительный опыт эксплуатации стенда подтвердил правильность методических и программных решений, использованных при создании АСКУ.

Все прикладное программное обеспечение для АСКУ тритиевых комплексов исследовательских установок ТРИТОН, АКУЛИНА и ПРОМЕТЕЙ было реализовано полностью в среде созданного автором пакета CRW-DAQ [19] и надежно работало под его управлением. Накопленный при создании АСКУ опыт и программный инструментарий могут использоваться для автоматизации других исследовательских установок в области тритиевых технологий [47,48,50,51].

Разработанный программный инструментарий нашел применение и в других исследовательских проектах. Например, при автоматизации системы управления системой охлаждения, измерений и стабилизации температуры фотонного спектрометра PHOS [20] в эксперименте ALICE (A Large Ion Collider Experiment) на Большом Адронном Коллайдере LHC (Large Hadron Collider) в ЦЕРН [21,22,23].

БЛАГОДАРНОСТИ

Автор выражает искреннюю благодарность научному руководителю д.ф.м.н. Ю.И. Виноградову за постоянное внимание, творческие советы, конструктивную критику и помощь, оказанную на всех этапах подготовки диссертационной работы.

Автор благодарен за многолетнее плодотворное сотрудничество своим коллегам и соавторам из РФЯЦ-ВНИИЭФ, ОИЯИ, НИИФ СПбГУ, СПбГУТ и ИТЭФ - С.В. Фильчагину, В.С. Арюткину, В.В. Травкину, А.А. Юхимчуку, С.К. Гришечкину, А.Н. Голубкову, А.М. Демину, С.В. Златоустовскому, Р.К. Мусяеву, Б.С. Лебедеву, В.В. Попову, В.В. Перевозчикову, П.В. Ширнину, В.Г. Зинову, В.В. Фильченкову, Д.Л. Демину, Н.Н. Графову, С.А. Юхимчуку, Г.М. Тер-Акопяну, А.М. Родину, А.С. Фомичеву, С.И. Сидорчуку, А.А. Курдюмову, И.Е. Габису, Е.А. Денисову, Е.А. Еварду, А.И. Лившицу, М.Е. Ноткину, А.А. Самарцеву, А.А. Буснюку, Л.Н. Богдановой.

Отдельную благодарность автор выражает Н.В. Завьялову и Ю.Ц. Оганесяну за интерес к работе и помощь в организационных вопросах.

Список использованных источников

1. А.А. Юхимчук, Р.И. Илькаев, В.Н. Лобанов. Изотопы водорода в научных программах РФЯЦ-ВНИИЭФ. // Материаловедение. 2002, № 2, С.35-42.
2. Л.И. Пономарев. Синтез и деление ядер как основа будущей энергетики. // Сборник докладов международного семинара "IHISM-04", 2005, С.15-22.
3. С.С. Герштейн, Ю.В. Петров, Л.И. Пономарев. Мюонный катализ и ядерный бридинг. // Успехи физических наук, 1990, Т. 160, Вып. 8, С.3–46.
4. В.М. Быстрицкий, В.П. Джелепов, В.И. Петрухин и др. Резонансная зависимость скорости образования мезомолекул $dd\mu$ в газообразном дейтерии // ЖЭТФ, 1979, Т. 76, Вып. 2, С. 460–469.
5. M.P. Faifman, L.I. Ponomarev. Resonant Formation of $dt\mu$ mesic molecules in the triple $H_2+D_2+T_2$ mixture // Physics Letters, 1991, Vol. 265, No. 3/4, P.201–206.
6. А.А.Юхимчук, В.А. Апасов, Ю.И. Виноградов и др. Комплекс газового обеспечения экспериментов по мюонному катализу ядерных реакций синтеза. // Приборы и техника эксперимента, 1999, №6, С.17-23.
7. А.А. Юхимчук. Разработка и создание комплекса для исследования мюонного катализа ядерных реакций синтеза в плотных смесях изотопов водорода. // Диссертация доктора технических наук. ОИЯИ, Дубна. 2002.
8. Ю.И. Виноградов, В.С. Арюткин, **А.В. Курякин** и др. Автоматизированная система контроля и управления комплексом подготовки газовой смеси для экспериментального исследования мюонного катализа ядерных реакций синтеза // Приборы и техника эксперимента, 2004, № 3. С.29–41.
9. Ю.И. Виноградов. Разработка и применение автоматизированных систем измерений, контроля и управления для исследований в области ядерной физики низких энергий. // Диссертация на соискание ученой степени доктора физико-математических наук. РФЯЦ-ВНИИЭФ, Саров, 2005.
10. A.M. Rodin, S.I. Sidorchuk, S.V. Stepantsov et al. High Resolution line for Secondary Radioactive Beams at the U400M Cyclotron // NIM B, 1997, Vol.126, P.236.
11. Yu.Ts. Oganessian, A.M. Rodin, **A.V. Kuryakin** et al. Status of ACCULINNA beam line. // NIM. B, 2003, Vol.204, P.114-118.

12. А.А. Юхимчук, Ю.Ц. Оганесян, Ю.И. Виноградов, **А.В. Курякин** и др. Система подачи изотопов водорода в ионный источник циклотрона У-400М. // Сборник докладов международного семинара "IHISM-04", 2005, С.401-409.
13. А.А. Yukhimchuk, Yu.I. Vinogradov, **A.V. Kuryakin** et al. Gas feeding system supplying the U-400M cyclotron ion source with hydrogen isotopes. // Fusion Science and Technology, July/August 2005, Vol.48, Num.1, P.704-707.
14. А.А. Юхимчук, М.Д. Вихарев, **А.В. Курякин** и др. Тритиевая мишень для исследования экзотических нейтронно-избыточных ядер и комплекс ее газового обеспечения. // ВАНТ, Серия: Физика ядерных реакторов, вып. ½, 2002, С.183-190.
15. А.А. Yukhimchuk, V.V. Perevozchikov, **A.V. Kuryakin** et al. Tritium target for research in exotic neutron-excess nuclei. // NIM A, 2003, V.513, No 3, P.439-447.
16. Ю.И. Виноградов, В.С. Арюткин, **А.В. Курякин** и др. Система контроля и управления комплекса тритиевой мишени для исследования экзотических нейтронно-избыточных ядер. // ВАНТ, Серия: Физика ядерных реакторов, вып. ½, 2002, С.197-200.
17. S.K.Grishechkin, A.N.Golubkov, E.V. Gornostaev et al. Device for investigations of tritium retention in and permeation through metals and structural materials. // Hydrogen Recycling at Plasma Facing Materials, 2000, Vol. 1, P.339-348.
18. Ю.И. Виноградов, **А.В. Курякин** и др. Автоматизированная система контроля, управления и сбора данных стенда "Прометей". // Материаловедение, 2002, №1, С.46-50.
19. **А.В. Курякин**, Ю.И. Виноградов. Программа для автоматизации физических измерений и экспериментальных установок (CRW-DAQ). // Свидетельство РФ об официальной регистрации программы для ЭВМ № 2006612848 от 10.08.2006 г.
20. V.I. Manko, **A.V. Kuryakin** et al. A high resolution electromagnetic calorimeter based on lead-tungstate crystals. // NIM A, 2005, V.550, Issues 1-2, P.169-184.
21. The ALICE Collaboration. The ALICE experiment at the CERN LHC. // 2008 JINST 3 S08002.
22. Lyndon Evans. The Large Hadron Collider. // New J. Phys., 2007, Vol.9, P.335.
23. ALICE Collaboration. First proton-proton collisions at the LHC as observed with The ALICE detector: measurement of the charged particle pseudorapidity density at $\sqrt{s} = 900\text{GeV}$. // The European Physical Journal C: Volume 65, Issue 1 (2010), Page 111.

24. А.А. Юхимчук, Ю.И. Виноградов, **А.В. Курякин** и др. Экспериментальное исследование реакции мюонного катализа dt в широком диапазоне условий D/T смеси. // Труды ВНИИЭФ, 2008 г., №12, стр.232-274.
25. V.R. Bom, A.M. Demin, **A.V. Kuryakin** et al. Experimental investigation of muon-catalyzed dt fusion in wide ranges of D/T mixture conditions. // ЖЭТФ, 2005, Том 127, Вып.4, С.752-779.
26. Yu.P. Averin, D.V. Balin et al. First measurements of $dt\mu$ cycle characteristics in liquid H/D/T mixture // Hyperfine Interactions, 1999, Vol.118, P.121–125.
27. D.L. Demin et al. The investigation of the dependence of the parameters of muon catalyzed fusion on the protium concentration in triple H/D/T mixture at high temperature and density. // Препринт ОИЯИ Е15-2000-156, Дубна, 2000.
28. V.R. Bom, V.V. Filchenkov, ..., **A.V.Kuryakin** et al. Measurement of the temperature dependence of the $dd\mu$ molecule formation rate in dense deuterium at temperatures 85-790 K // ЖЭТФ, 2003, Т.123, №. 3, С.518–526.
29. V.R. Bom et al. Investigation of the parameters of muon catalyzed fusion in double D/T mixture at high temperature and density. // Abstracts of the International RIKEN Conference. Muon catalyzed fusion and related exotic atoms. April 22-26, 2001. Shimoda, Japan. P. 3.
30. Yu.P. Averin et al. Experimental results on the muon catalyzed dt fusion in H/D/T mixture. // Abstracts of the International RIKEN Conference. Muon catalyzed fusion and related exotic atoms. April 22-26, 2001. Shimoda, Japan. P. 3–8.
31. V.V. Filchenkov, N.N. Grafov. Status of the investigation of the muon catalyzed $d+t$ reaction. // Prepr. of JINR E15-2003-96, Dubna, 2003.
32. L.N.Bogdanova, V.R.Bom, **A.V.Kuryakin** et al. Search for the radioactive capture $d+d \rightarrow {}^4\text{He}+\gamma$ reaction from the $dd\mu$ muonic molecule state. // ЯФ, 2002, Том 65, №10, С.1826-1832.
33. V.R. Bom, L.N. Bradbury, J.D. Davies et al. Temperature and density dependence of μ -catalysis cycling rate in dense D/T and H/D/T gas mixtures. // Hyperfine Interactions, 1999, Vol.118, P.103–109.
34. Yu.Ts. Oganessian, A.M. Rodin, **A.V. Kuryakin** et al. Evidences for resonance states in ${}^5\text{H}$. // Physics Letters B, 2003, V.556, P.70-75.

35. G.M. Ter-Akopian, A.S. Fomichev, M.S. Golovkov, Yu.I. Vinogradov et al. Recent study of the ${}^3\text{H}(t,p)$ reaction induced by 58-MeV tritons. Evidences for excited states in ${}^5\text{H}$. // Abst. of International Symposium on Exotic Nuclei EXON-2004, Peterhof, Lake Ladoga, July 5-12, 2004. JINR, Dubna, 2004. P.29.
36. M.S. Golovkov, L.V. Grigorenko, A.S. Fomichev, Yu.I. Vinogradov et al. Observation of Excited States in ${}^5\text{H}$ // Physical Review Letters, 2004, Vol.93, P.262-501(1-4).
37. Ю.Ц. Оганесян, Г.М. Тер-Акопян, Ю.И. Виноградов, **А.В. Курякин** и др. Изучение структуры ультранейтронно-избыточных ядер водорода и гелия с использованием реакций радиоактивных пучков на тритиевой мишени. // Известия РАН, Сер. Физическая, 2002, Т. 66, № 5, С.619–624.
38. M.S. Golovkov, Yu.Ts. Oganessian, G.M. Ter-Akopian, Yu.I. Vinogradov et al. Super-heavy hydrogen isotopes studied with 58 MeV triton beam. // Proc. of the Int. Simp. on Exotic Nuclei, Lake Baikal, Russia, July 24-28, 2001. P. 334-346.
39. С.И. Сидорчук. Экспериментальное исследование изотопов водорода ${}^{4,5,7}\text{H}$ в реакциях на пучках ядер ${}^3\text{H}$, ${}^8\text{He}$. // Диссертация кандидата физ.-мат. наук. ОИЯИ. Дубна. 2004.
40. Р.К. Мусяев, Е.А. Денисов, А.А. Юхимчук и др. Исследования явлений проницаемости и сверхпроницаемости металлов на установке "Прометей". // Сборник докладов международного семинара "IHISM-01", 2002, С.228-233.
41. Р.К. Мусяев, Б.С. Лебедев, А.А. Юхимчук и др. Сверхпроницаемость трития: экспериментальное исследование и моделирование рециркуляции трития на установке "Прометей". // Сборник докладов международного семинара "IHISM-04", 2005, С.84-91.
42. R.K. Musyaev, B.S. Lebedev, S.K. Grischechkin, A.A. Yukhimchuk et al. Tritium superpermeability: experimental investigation and simulation of tritium recirculation in "Prometheus" setup. // Fusion Sci. Technol., 2005, Vol.48, P.35-38.
43. Р.К. Мусяев, Б.С. Лебедев, А.А. Юхимчук и др. Исследование явления сверхпроницаемости изотопов водорода через ванадиевую мембрану на установке "Прометей". // Сборник докладов международного семинара "IHISM-07", 2008, С.165-169.

44. А.И. Лившиц, А.А. Юхимчук, Р.К. Мусьяев и др. Сверхпроницаемость водорода в металлах V группы — применение для откачки и рециркуляции топливной смеси в термоядерных реакторах. // *Материаловедение*, 2005, №8, С.40.
45. R.K.Musyaev, A.A. Yukhimchuk, B.S. Lebedev et al. Study of hydrogen isotopes superpermeation through vanadium membrane on Prometheus setup. // *Fusion Science and Technology*, 2008, Vol.54, P.523-525.
46. И.Л. Малков, С.К. Гришечкин, А.А. Юхимчук и др. Получение образцов, содержащих ^3He , для исследования физико-механических свойств конструкционных материалов. // *Сборник докладов международного семинара "IHISM-07"*, 2008, С.78-89.
47. **А.В. Курякин**, Ю.И.Виноградов. Программное обеспечение для автоматизации исследовательских установок. // *Сборник докладов международного семинара "IHISM-04"*, 2005, С.411-419.
48. Yu.I.Vinogradov, **A.V. Kuryakin**, A.A. Yukhimchuk. Measurement and control systems of tritium facilities for scientific research. // *Fusion Science and Technology*, July/August 2005, Vol.48, Num.1, P.696-699.
49. Yu.I. Vinogradov, A.V. Kuryakin. The basic software for "TRITON" and "ACULINA" - the automated tritium complexes. // *Proceedings of the International Conference of Muon Catalyzed Fusion and Related Topics (μCF-07)*. June 18-21, 2007. Dubna, P. 361-368.
50. **А.В. Курякин**, Ю.И. Виноградов. Программное обеспечение автоматизированных измерительных систем в области тритиевых технологий. // *ВАНТ, серия «Термоядерный синтез»*, 2008 г., выпуск 2, стр. 80-90.
51. Ю.И. Виноградов, **А.В. Курякин**, А.А.Юхимчук. Системы измерений и управления тритиевых установок для научных исследований. // *Изотопы водорода. Фундаментальные и прикладные исследования. Сборник статей*. 2009 г., Саров, стр. 678-681.
52. D.L. Demin, V.P. Dzheleпов et al. *Hyperfine Interaction* 101/102(1996)583.
53. Н.Н. Графов, В.Г. Гребинник, Д.Л. Демин и др. Жидкотритиевая мишень. // *Приборы и техника эксперимента*, 1999, №1, С.21-27.
54. В.В. Перевозчиков, А.А. Юхимчук, Н.С. Ганчук и др. Тритиевая мишень высокого давления. // *Приборы и техника эксперимента*, 1999, №1, с.28-33.

55. V.V. Perevozchikov, A.A. Yukhimchuk, Yu.I. Vinogradov et al. Deuterium high pressure target. // Preprint of the JINR, D13-2001-118, Dubna, 2001.
56. П.В. Ширнин, Ю.И.Виноградов, **А.В. Курякин** и др. Количественный анализ изотопов водорода газовым радиохроматографом РХТ-2. // Препринт ВНИИЭФ 83-2002.
57. С.В. Фильчагин, Ю.И. Виноградов, **А.В. Курякин** и др. Автоматизация радиохроматографа РХТ-3 для изотопного и молекулярного анализа водородосодержащих газовых смесей. // ВАНТ, серия «Термоядерный синтез», 2008, Вып.2, С.93-97.
58. Ю.И. Виноградов, С.М. Придчин, **А.В. Курякин** и др. Многоканальная автоматизированная система радиационного мониторинга // Материаловедение. 2002, № 6. С. 53–55.
59. Г. Олссон, Д. Пиани. Цифровые системы автоматизации и управления. // изд. Невский диалект, Ст. Петербург, 2001 г.
60. П. Дарахвелидзе, Е. Марков. Delphi - среда визуального программирования. // изд. ВНУ, Санкт-Петербург, 1996 г.
61. Е.В. Шикин, А.И. Плис. Кривые и поверхности на экране компьютера. // изд. ДИАЛОГ-МИФИ, Москва, 1996 г.
62. В.В. Носач. Решение задач аппроксимации с помощью персональных компьютеров. // изд. Бином, 1994 г.
63. Хемминг. Цифровые фильтры. // изд. Недра, Москва, 1989 г.
64. Прикладное непараметрическое сглаживание. // Москва, 1993 г.
65. Дж. Форсайт, М. Малькольм, К. Моулер. Машинные методы математических вычислений. // изд. Мир, Москва, 1980 г.
66. Д.М. Химмельблау. Прикладное нелинейное программирование. // изд. Мир, Москва, 1975 г.
67. Wirth N. The programming language Pascal. // Acta Informatica, 1971, Vol. 1, P. 35—63.
68. Р. Кёртен. Введение в QNX Neutrino 2. Руководство для разработчиков приложений реального времени. // изд. БХВ-Петербург, 2005 г.
69. Joe Armstrong. Making reliable distributed systems in the presence of software errors. // A Dissertation of Doctor of Technology, The Royal Institute of Technology Stockholm, Sweden, December 2003. www.sics.se/~joe/thesis/.

70. C. Gaspar, M. Donzelmann. DIM - A Distributed Information Management System for the Delphi experiment at CERN. // IEEE Eight Conference REAL TIME '93 on Computer Applications in Nuclear, Particle and Plasma Physics (Vancouver, June 8-11 1993) , <http://dim.web.cern.ch/>.
71. А.А. Самарцев. Проникновение водорода сквозь металлические мембраны в области низких давлений. // Кинетика и катализ, 1985, Vol.24, P.815.
72. A.I. Livshits, M.E. Notkin, Yu.M. Pustovoi, A.A. Samartsev. Superpermeability of solid membranes and gas evacuation. // Vacuum, 1979, Vol.29, P.113.
73. A.I. Livshits, M.E. Notkin and A.A. Samartsev. Physico-chemical origin of superpermeability — large-scale effects of surface chemistry on "hot" hydrogen permeation and absorption in metals. // J.Nucl.Mater., 1990, Vol.170, P.74.
74. В.В. Попов, И.Е. Габис, Ю.И. Заика. Исследование водородопроницаемости методом концентрационных импульсов: математическое моделирование. // Сборник докладов международного семинара "IHISM-04", 2005, С.35-44, г.
75. И.Е. Габис. Метод концентрационных импульсов для исследования транспорта водорода в твердых телах. // ЖТФ, 1999, Т.69, С.99-103.
76. A.N. Golubkov, A.A. Yukhimchuk, Journal of the Moscow physical society, Vol. 9, № 3, 1999, p. 223.
77. В.Г. Клевцов, В.Н. Лобанов, Ю.А. Хабаров, А.А. Юхимчук. Диффузионная очистка изотопов водорода палладиевыми фильтрами. // Сборник докладов международного семинара "IHISM-01", 2002, с.136–142.
78. T. Matsuzaki, K. Nagamine, M. Tanase et al. // MCF, 1999, Vol.119, P.361.
79. I. Tilquin, Y.E. Masri, M. Parlog et al. Detection Efficiency of the Neutron Modulator Detector DEMON and Related Characteristics // NIM A. 1995. Vol. 365. No. 2/3. P. 446–461.
80. М.М. Соколов. Возможность применения измерительно-вычислительного комплекса Токамака-10 на других экспериментальных установках. // ВАНТ. Сер. «Термоядерный Синтез», 2003, Вып.4, С.73–78.
81. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. // Пер. с англ.: М.: Издательство ТРИУМФ, 2002 – 816 с.

Приложение. Особенности реализации инструментального пакета CRW-DAQ

1. Системы реального времени и отказоустойчивые системы

Программный пакет CRW-DAQ создан для разработки систем реального времени (СРВ), какими являются АСКУ для исследовательских установок, работающих в реальном времени (РВ). Версия CRW16 под DOS/DPMI может работать как система жесткого РВ с реакцией ~10 мкс, а CRW32 под Windows-NT – как система мягкого РВ с реакцией ~10 мс. Понятие СРВ вызывает много споров, поэтому здесь уместно дать определение и способы реализации СРВ.

Каноническое определение СРВ дано канадским математиком Дональдом Гиллиесом (Donald Gillies, www.faqs.org/faqs/realtime-computing/faq/): «Системой реального времени является такая система, корректность функционирования которой определяется не только логической корректностью выполнения вычислений, но и временем (реакции), в которое получен требуемый результат. Если требования по времени не выполняются, то считается, что произошел отказ системы». Толковый словарь по информатике (Першиков В.И., Савинков В.М. М.: Финансы и статистика, 1991) дает следующее определение СРВ: «Режим реального времени (real time processing) - режим обработки данных, при котором обеспечивается взаимодействие вычислительной системы с внешними по отношению к ней процессами в темпе, соизмеримом со скоростью протекания этих процессов». Этих определений достаточно, чтобы показать главные свойства СРВ.

Всякая СРВ связана с некоторым «внешним по отношению к ней» объектом управления, свойства которого и определяют требуемое от неё время реакции. Чисто вычислительная система, не связанная с каким-либо внешним объектом, не является СРВ - корректность ее работы не зависит от времени расчетов. Далее, определение ничего не говорит о конкретной величине времени реакции СРВ – требуется лишь, чтобы оно соответствовало объекту управления. Об этом не следует забывать, потому что нет смысла бороться за очень малое время реакции системы управления, если объект управления этого не требует. Незачем, например, регулировать температуру паяльника с частотой 100 кГц – все равно температура

так быстро изменяться не сможет. В то же время слишком завышенные требования по времени реакции могут сделать систему дорогой и неустойчивой.

Не следует путать СРВ с высокопроизводительными системами – средняя или интегральная производительность не имеет прямого отношения к РВ. Не надо также называть СРВ системы быстрой регистрации – они ничем не управляют (нет объекта управления), поэтому в них нет понятия времени реакции. СРВ отличаются не высокой (в среднем) производительностью – они могут быть реализованы на относительно медленной технике. Они не требуют высокой (в среднем) скорости регистрации событий – она легко обеспечивается аппаратной буферизацией. СРВ отличается в первую очередь своевременной реакцией на асинхронные (по отношению к системе управления) события, которую обеспечивает в первую очередь архитектура СРВ, а не высокая скорость работы отдельных компонентов (процессора, памяти, дисков и т.д.) Это особенно важно потому, что серийная аппаратура и ПО ориентированы чаще всего именно на обеспечение высокой средней производительности, а не малого времени реакции. Не надо также путать СРВ с интерактивными системами, работающими в online режиме – для СРВ важна своевременность реакции не на действия пользователя, а на события в объекте управления (даже в ущерб интерактивности).

Следует отметить ряд менее очевидных особенностей СРВ. Из определения Гиллиеса следует тесная связь СРВ и отказоустойчивых систем – задержка реакции считается отказом системы и должна рассматриваться наравне с другими вероятными отказами (сбой питания, отказ аппаратуры, ошибка программы и т.д.) Поэтому ряд методов обеспечения отказоустойчивости (например, избыточность или дублирование управления) могут применяться и для создания СРВ.

Далее, понятие СРВ всегда относится к аппаратно-программным комплексам, поскольку СРВ всегда связана с внешним объектом, а следовательно с регистрирующей аппаратурой. Нельзя говорить о СРВ по отношению к аппаратуре или к ПО (например, ОС) отдельно друг от друга – СРВ может быть только система в целом. Нет, например, смысла улучшать ПО, если аппаратура не позволяет строить СРВ. С другой стороны, отсюда также следует вывод, что, аппаратная часть в ряде случаев может компенсировать недостатки ПО, и наоборот.

Центральным понятием СВВ является её время реакции. В структуру любой СВВ (Рис.58,1) входит объект управления - именно он (а не компьютер!) определяет требуемое время реакции. Измерительный модуль (АЦП, входной регистр и т.д.) регистрирует событие (Рис.58,2) в объекте управления. Решающий модуль (логическая схема, контроллер, промышленный компьютер и т.д.) вырабатывает реакцию, то есть принимает решение о том, что надо делать в ответ на принятое событие. Исполнительный модуль (ЦАП, выходной регистр и т.д.) реализуют реакцию, то есть исполняют принятое решение. Время реакции СВВ определяется как суммарное время прохождения всей цепочки, от возникновения события до исполнения реакции, выработанной системой управления.

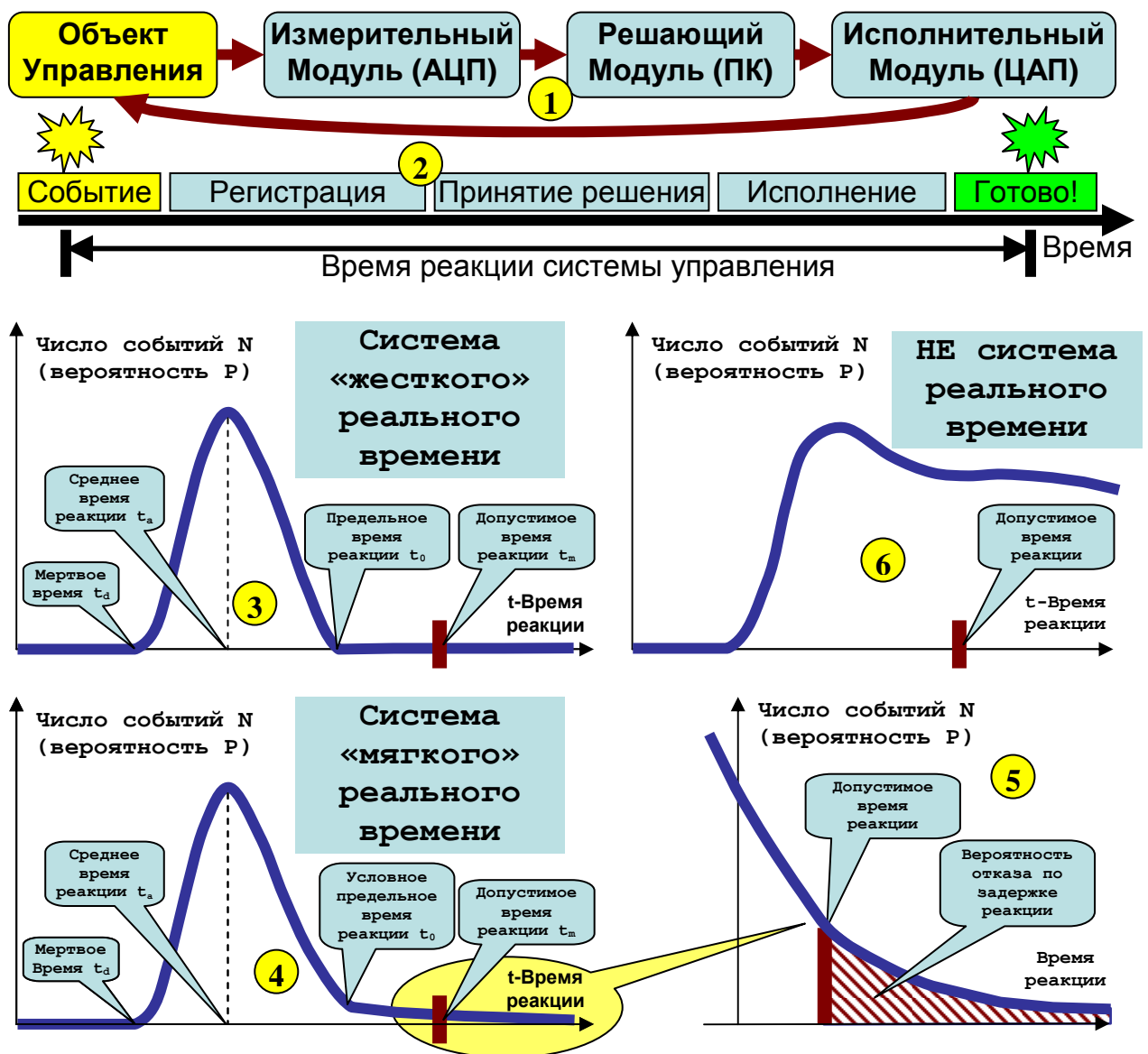


Рис.58. Структура системы реального времени (1), время реакции СВВ (2), гистограммы систем «жесткого» (3) и «мягкого» (4) РВ и системы, не работающей в реальном времени (5), а также определение вероятности отказа по задержке реакции.

Наиболее наглядное описание свойств СРВ дает гистограмма распределения числа событий $N(t)$ или плотности вероятности $P(t)$ по времени реакции t . Это распределение имеет ряд особенностей (Рис.58,3). Во-первых, всякая СРВ имеет «мертвое» время t_d , поскольку регистрация и исполнение требуют времени (которое зависит от аппаратуры), даже если решение принимается мгновенно. Далее, СРВ имеет некоторое характерное среднее время реакции t_a на большинство событий и практически всегда имеет некий разброс по времени реакции. Этот разброс неизбежен в современных компьютерных системах, так как они очень сложны и имеют ряд элементов с недетерминированным временем исполнения. Например, современные процессоры не имеют фиксированного времени исполнения инструкций из-за наличия кэша, общей шины доступа к памяти, спекулятивного исполнения команд, предсказания ветвлений, динамического управления частотой тактов по датчику температуры ядра или при режиме энергосбережения, а также из-за необходимости доступа к ряду аппаратных ресурсов общего пользования, что приводит к циклам ожидания освобождения этих ресурсов. Все эти технологии нацелены на повышение средней производительности ПК, однако вносят во временные свойства системы существенную неопределенность (индетерминизм).

СРВ делят также на системы «жесткого» и «мягкого» РВ. Под жесткими СРВ понимают системы, в которых задержка реакции недопустима ни в коем случае (приводит к серьезной аварии), под мягкими СРВ – системы, в которых задержка реакции нежелательна, но допустима (она снижает качество обслуживания, но не ведет к аварии). Другими словами «жесткие» СРВ «всегда успевают», а «мягкие» СРВ – «обычно успевают».

Для систем жесткого РВ (Рис.58,3) характерно наличие предельного времени реакции t_0 , после которого вероятность событий равна точному нулю: $P(t > t_0) = 0$. Если предельное время реакции не превышает допустимого (для данного объекта управления) времени t_m , то СРВ работает безотказно.

Для систем мягкого РВ (Рис.58,4) характерно наличие условного предельного времени реакции t_0 , после которого вероятность событий резко падает: $P(t > t_0) \ll 1$. Однако оно не равно точному нулю: $P(t > t_0) \neq 0$. Даже если условное предельное время реакции не превышает допустимого (для данного объекта

управления) времени t_m , то СРВ допускает некоторую вероятность отказа, которая определяется «хвостом» (Рис.58,5) распределения $\int_{t_m}^{\infty} P(t) \cdot dt$.

Наконец, система управления не может быть названа СРВ (Рис.58,6), если число событий не имеет пренебрежимо малого значения при временах выше допустимого времени реакции t_m , то есть $P(t > t_m) > 0$.

Из сказанного следует, что жесткие СРВ имеют точное, формальное определение, в то время как мягкие СРВ определены приблизительно, условно. Существует пуританский соблазн отказать системам мягкого РВ в праве на существование - на том основании, что они допускают вероятность отказа по задержке реакции. Этому мешает ряд обстоятельств. Во-первых, не во всех случаях отказ по задержке означает аварию – это может быть повышение стоимости, снижение эффективности, то есть нечто нежелательное, но не опасное. Допустить возможность отказа по задержке в этом случае можно, если стоимость потерь из-за отказов не превышает стоимости усилий на обеспечение режима жесткого РВ.

Другой аргумент более существенный. Он состоит в том, что абсолютно жесткие системы РВ, как и абсолютно безотказные системы, являются абстракцией, недостижимым идеалом. Отказоустойчивая система (и СРВ в том числе) должна строиться на том допущении, что вероятность отказа (в том числе по задержке реакции) существует всегда. Тем более что теоретически доказать существование точного предельного времени реакции бывает довольно трудно (например, из-за отмеченного выше временного индетерминизма современных компьютерных систем). Условность и неточность определения мягких СРВ исчезнет, как только мы станем рассматривать задержку реакции в терминах хорошо разработанной теории надежности, как один из возможных источников сбоев, наравне с программными ошибками, сбоями аппаратуры или питания.

Рассмотрение СРВ в контексте отказоустойчивых систем – не простая игра словами, это вполне конструктивный подход. Отказоустойчивые системы строятся по определенной методике, пригодной, в том числе, для построения СРВ. При этом ставится задача создания безотказной системы при условии, что каждый из отдельных компонентов системы допускает вероятность отказа.

Одним из главных методов построения отказоустойчивых систем является введение избыточности (аппаратной, информационной, временной). Так, в канале регулирования температуры нагрева источников водорода (раздел 3.3.2) введение дублирующего элемента управления (электромагнитного реле) позволяет резко снизить опасность аварии из-за возможного пробоя семистора в электронном регуляторе – в этом случае цепь питания нагревателей будет коммутировать реле.

Аналогично может решаться и проблема сбоев по задержке реакции в СРВ. Мягкая СРВ допускает вероятность отказа по задержке? Жесткая СРВ недоступна? Прекрасно, давайте примем меры для обеспечения отказоустойчивости системы, введем дублирующий элемент управления, корректирующий последствия обнаруженного отказа по задержке реакции в основной системе управления.

Наиболее распространенным способом такого дублирования являются сторожевые таймеры (watchdog). Например, все модули серии I-7000 имеют аппаратные сторожевые таймеры (с порогом от 100 мс). Более того, наличие watchdog было одним из существенных аргументов при выборе аппаратуры.

Система управления, снабженная watchdog таймером, работает следующим образом. В нормальной ситуации работает главная система управления, обычно контролируемая компьютером. В ее задачи входит управление объектом (оно может быть достаточно сложным) и периодический сброс watchdog таймера, который информирует его о том, что система управления контролирует ситуацию. Если главная система управления вовремя не сбросила watchdog, то он считает, что произошел отказ системы (по задержке реакции, по разрыву связи - неважно) и принимает управление на себя. Обычно watchdog таймеры достаточно просты и просто отключают контролируемую ими аппаратуру или переводят её в некоторое заранее определенное безопасное состояние. Выход из этого состояния требует специального сигнала от главной системы управления, после которого watchdog возвращает ей управление. Следствием введения watchdog таймера является то, что задержка реакции мягкой СРВ перестает быть аварийно опасным отказом и переходит в разряд неблагоприятных факторов, снижающих качество управления, но не представляющих опасности. Кроме того, watchdog переводит мягкую СРВ в класс жестких СРВ, так как теперь становится возможным точно указать предельное время реакции – его, в худшем случае, гарантирует watchdog.

Именно по такому принципу построены АСКУ установок ТРИТОН, АКУЛИНА, ПРОМЕТЕЙ и большинство других АСКУ, работающих под управлением пакета CRW-DAQ. Как упомянуто в разделах 1.4.2.7, 3.3.2, управление в этих системах реализовано в основном с помощью модулей I-7000, снабженных watchdog таймером, который переводит аппаратуру в безопасное состояние по срабатыванию таймера. Это позволяет строить высоконадежные, отказоустойчивые СРВ, несмотря на то, что ОС Windows-NT/2000/XP/VISTA/7, под управлением которой работает CRW-DAQ, не является системой жесткого РВ.

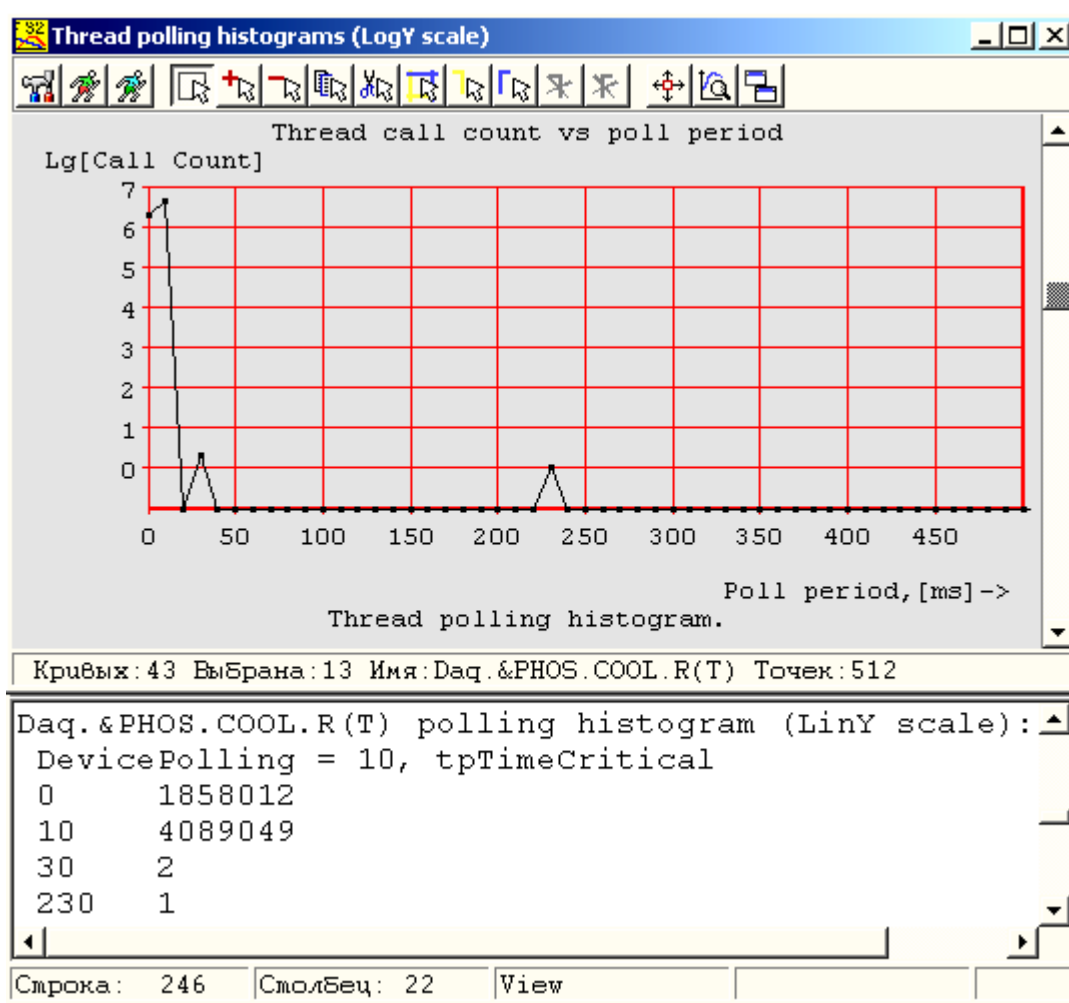


Рис.59. Типичная гистограмма (в логарифмической шкале) периода опроса управляющего программного потока. Большинство опросов ($\sim 5 \cdot 10^6$) выполнено с периодом ≤ 10 мс, однако есть 2 события с задержкой 30 мс и 1 событие с задержкой 230 мс. Это и есть «хвост» распределения $N(t)$.

Относительно временных характеристик CRW-DAQ можно сказать следующее. Версия CRW16 под DOS/DPMI позволяет выполнять обработку аппаратных прерываний, при этом можно добиться жесткого времени реакции около 10 мкс (время обработки прерывания). Однако DOS – устаревшая система, и к тому же на прерываниях можно построить лишь достаточно простые системы управления, а режима многозадачности под DOS нет. Поэтому большинство АСКУ

работают под управлением версии CRW32 пакета под Windows-NT/XP/VISTA/7, где многозадачность поддерживается, несмотря на то, при этом жесткого РВ нет.

Версия CRW32 является мягкой СРВ и (как большинство мягких СРВ) предполагает периодический (по таймеру ПК) программный опрос аппаратных устройств. Ядро Windows NT позволяет организовать периодический опрос со средним временем опроса от 1 до 10 мс, при этом повышение приоритета потока позволяет повысить стабильность частоты его опроса, хотя и не дает полной гарантии своевременного опроса. В отличие от других мягких СРВ, пакет CRW-DAQ имеет контроль и диагностику периода опроса (Рис.59) всех программных потоков, то есть времени между последовательными вызовами прикладной программы по часам компьютера. Хотя период опроса не является в точном смысле временем реакции системы, но он с ним явно связан: реакция системы управления составляет 2-3 периода опроса, с учетом взаимодействия нескольких параллельных потоков и аппаратной части для решения общей задачи. Ядро пакета CRW32 постоянно гистограммирует период опроса всех программных потоков для их последующего анализа, а также имеет программный сторожевой таймер для диагностики повисших потоков. На примере типичной гистограммы (Рис.59) опроса управляющего потока видно, что подавляющее большинство (в данном случае $\sim 5 \cdot 10^6$) опросов прикладного потока произведено с периодом ≤ 10 мс, однако есть три события (два по 30 мс и одно 230 мс) с задержкой реакции, превышающей заданный период опроса (10 мс). Это и есть «хвост» распределения $N(t)$ (Рис.58,5), который приходится корректировать за счет дублирования управления с помощью watchdog таймера.

Сказанное можно резюмировать так: пакет CRW-DAQ позволяет строить мягкие СРВ со временем реакции ~ 10 мс, а также жесткие отказоустойчивые СРВ с реакцией ~ 100 мс при условии использования аппаратной части, снабженной watchdog таймером. При этом возможно временное снижение качества управления, связанное с вероятностью задержки реакции со стороны ПК и срабатывания сторожевого таймера, выполняющего лишь самое простое управление.

2. Примеры калибровок

Ниже приведено два примера, иллюстрирующих применение аппарата калибровки (раздел 1.4.2.3), а также формата файлов калибровки.

Пример 1: Калибровка линейного канала.

Если датчик имеет слабую нелинейность, можно искать зависимость физической величины y от сигнала x в виде линейной функции $y = c_0 + c_1x$. Для этого в терминах калибровок CRW-DAQ мы должны положить

$$f_x(x, z) = x, f_x^{-1}(x', z) = x', f_y(y, z) = y, f_y^{-1}(y', z) = y' \quad (40)$$

$$n = 1, x_0 = 0, \Delta = 1 \quad (41)$$

Пусть, например, x - милливольты АЦП [mV], y - давление в Паскалях [Pa].

Тогда описание калибровки в пакете CRW-DAQ будет иметь примерно такой вид:

```
[mV-Pa calibration]
FitMethod = Polynom
TransformX = Line
TransformY = Line
Power      = 1
Center     = 0
Scale      = 1
Data      mV      Pa      Weight
          0.065    0       1
          21.210   1383    1
          40.203   2624    1
End Data
```

Здесь TransformX, TransformY задают тождественное преобразование (40), Power, Center, Scale - параметры (n, x_0, Δ) (41), а секция Data задает массив точек калибровки, взятых с весами Weight.

Пример 2: Калибровка термопарного канала.

Пусть x - код АЦП термопары [mV], y - температура термопары [$^{\circ}$ C], z - температура холодного спая [$^{\circ}$ C]. Зависимость напряжения y' «идеальной» термопары от ее температуры y и температуры холодного спая z записывается как

$$y' = E(y) - E(z) \quad (42)$$

где $E(t)$ - стандартная термопарная зависимость, построенная по таблицам ГОСТ. Для интерполяции этой таблицы используется сглаживающий сплайн (10).

Зависимость напряжения термопары y' от кода АЦП x линейна и определяется усилением измерительного канала, то есть описывается полиномом $y' = c_0 + c_1x'$. Поэтому функции линеаризации могут быть записаны в виде:

$$f_x(x, z) = x, f_x^{-1}(x', z) = x', f_y(y, z) = E(y) - E(z), f_y^{-1}(y', z) = E^{-1}(y' + E(z)) \quad (43)$$

где $E(t)$ - термопарная таблица, $E^{-1}(t)$ - обратная к ней функция. В новых линеаризованных переменных зависимость $y'(x')$ описывается полиномом

$p_{n,c,x_0,\Delta}(x')$ с параметрами $n = 1$, $x_0 = 0$, $\Delta = 1$. Когда оптимальные коэффициенты c^* полинома найдены, зависимость $y(x)$ записывается как

$$y(x, z) = E^{-1}(p_{n,c^*,x_0,\Delta}(x) + E(z)) \quad (44)$$

Описание этой калибровки в калибровочном файле имеет примерно такой вид:

```
[U(mV)-T calibration]
FitMethod = Polynom
TransformX = Line
TransformY = HRCP
Power = 1
Center = 0
Scale = 1
Data U(mV)      T      Weight      Tc
      -9.724    -209      1          0
           0         0         1          0
      67.222     809      1          0
End Data
Notice Text
      Стандартная калибровка хромель-копель
      ГОСТ Р 50431-92 (МЭК 584-1-77)
End Notice Text
```

Отметим, что хранение в файлах исходных данных калибровки вместо калибровочных коэффициентов позволяет впоследствии легко изменять параметры калибровки (степень полинома, тип шкалы и т.д.) без потери точности. Кроме того, файлы калибровки тем самым документируют калибровочные измерения (часто дорогостоящие) и позволяют точно зафиксировать их условия, что важно для проверки достоверности измерений при обработке данных эксперимента.

3. Интерпретатор языка DaqScript

Одним из инструментов offline обработки данных в пакете CRW-DAQ являются программы на С-подобном языке DAQ Script (раздел 1.4.2.5), работающие подобно динамическим расширениям DPE (Рис.61), но вместо компиляции и загрузки DLL создается экземпляр интерпретатора, непосредственно исполняющий код. Несмотря на замедление обработки при интерпретации, его использование оправдано повышением отказоустойчивости программ, так как интерпретатор хорошо защищен от ошибок в прикладных программах.

Интерпретатор DaqScript разрабатывался как основа для командного языка главной консоли пакета CRW-DAQ (Рис.60.1). Позже он нашел другие применения.

Интерпретатор имеет один тип данных – вещественный (double), целые числа считаются его подмножеством. Строки, хотя и не являются типом данных, могут служить аргументом команд. Кроме арифметических выражений, включающих операторы +, -, *, /, %, = и скобки (), поддерживаются переменные,

встроенные константы, функции и команды. Переменные объявляются оператором «*var*», а также создаются и уничтожаются оператором присвоения «*=*». Так, выражения «*x=1*», «*var x*» создают переменную *x*, а пустое присвоение «*x=*» уничтожает ее. Функции отличаются скобками после имени, например, «*sin(x)*», а команды распознаются по первому символу «*@*», например, «*@echo Hello, World*». Остаток строки после имени команды считается ее аргументом и позволяет интерпретировать строковые данные. В интерпретаторе имеются также условный оператор «*if(...) then ...*», символьные метки, оператор перехода к метке «*goto ...*», вызова метки «*gosub ...*» и возврата «*return*», что позволяет создавать несложные программы, именуемые макросами. Отметим, что метки заканчиваются двоеточием «*:*», а комментарии отделяются точкой с запятой «*;*».

Поскольку DAQ Script был задуман как командный язык для консольного интерпретатора, он носит директивный характер, и в нем нет многих привычных конструкций процедурного программирования. Так, в нем есть встроенные функции и команды, определяемые при компиляции пакета, но нет декларации новых функций и команд при интерпретации. Нет также операторов цикла, которые, однако, можно заменить комбинацией операторов «*if*», «*gosub*», «*goto*». Так, для вывода в консоль ряда чисел можно написать макрос:

```
i=0
Loop:
  if(ge(i,10)) then goto Quit
  gosub PrintNumber
  i=i+1
  goto Loop
PrintNumber:
  @echo "i" value is %i%
  return
Quit: exit
```

Приведем примеры применения DaqScript (Рис.60). Он используется как командный язык в главной консоли (1) и реализует много команд, недоступных в графическом режиме, причем все эти команды может инициировать как оператор в консоли, так и прикладная программа АСКУ, написанная на языке DAQ Pascal. Он используется в интерактивном калькуляторе формульных выражений (2), в диалоговом графопостроителе (3), позволяющем строить графики заданных формулами функций (4). Он используется в макросах окон кривых (раздел 1.4.2) для первичной обработки экспериментальных данных, сглаживания и фильтрации. Наконец, DaqScript работает в прикладных программах АСКУ, причем каждый

интерпретатор имеет свой программный поток, который периодически (с периодом $1 \div 5000$ мс) вызывает макрос DaqScript, выполняющий сбор или обработку данных.

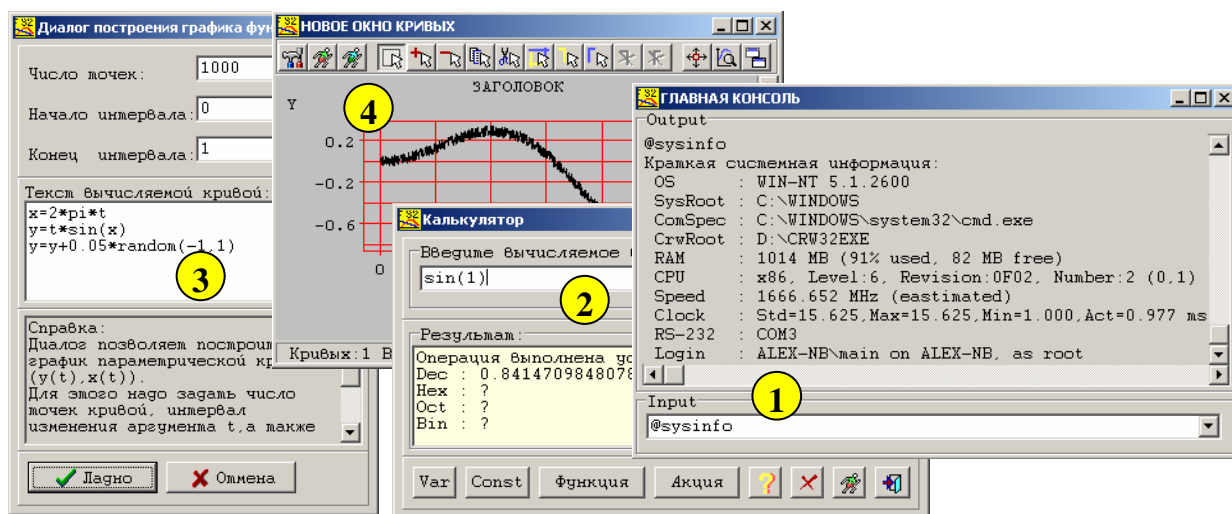


Рис.60. Применения DAQ Script: консоль, калькулятор, графопостроитель.

Хотя интерпретатор не отличается высокой скоростью исполнения ($\sim 10^5$ операторов в секунду), но он незаменим, когда вычисляемое выражение неизвестно в момент создания системы, а вводится оператором интерактивно. Кроме того, он хорошо защищен от ошибок в прикладном коде, что повышает надежность систем управления. Наконец, в интерпретатор легко добавлять новые встроенные функции без снижения уровня защищенности пакета, что позволяет резко сократить затраты на его разработку. Поэтому многие функции пакета, не требующие высокой скорости исполнения, реализованы как команды DaqScript.

Отметим, что идея создания DaqScript родилась при работе с программой Expression Evaluator (Mark Morley, 1992) на языке C. Реализация интерпретатора на языке Pascal и его дальнейшее развитие полностью принадлежит автору.

4. Реализация динамических расширений DPE

Наиболее мощным средством обработки данных в пакете CRW-DAQ являются динамические расширения DPE (Dynamic Plug-in Extension). Имеется библиотека готовых DPE расширений, но главное, что эти расширения могут создаваться и редактироваться в пакете в любое время, не требуя компиляции всего пакета и позволяя расширять его функции по мере потребности.

Обычно фазы разработки и исполнения программ явно разделены (Рис.61,а). Исходный код программы PAS на языке PASCAL редактируется, компилируется в исполняемый EXE файл и запускается на выполнение. Запускаемая программа

считывает входные данные и по команде оператора вызывает процедуру обработки, возвращающую ее результат. У этой простой схемы есть недостатки. При изменении процедуры обработки нужно завершать программу, компилировать и запускать снова, многократно повторяя запуск программы, ввод данных и команды оператора для вызова процедуры обработки. При этом «под рукой» необходимо иметь компилятор, что не всегда доступно.

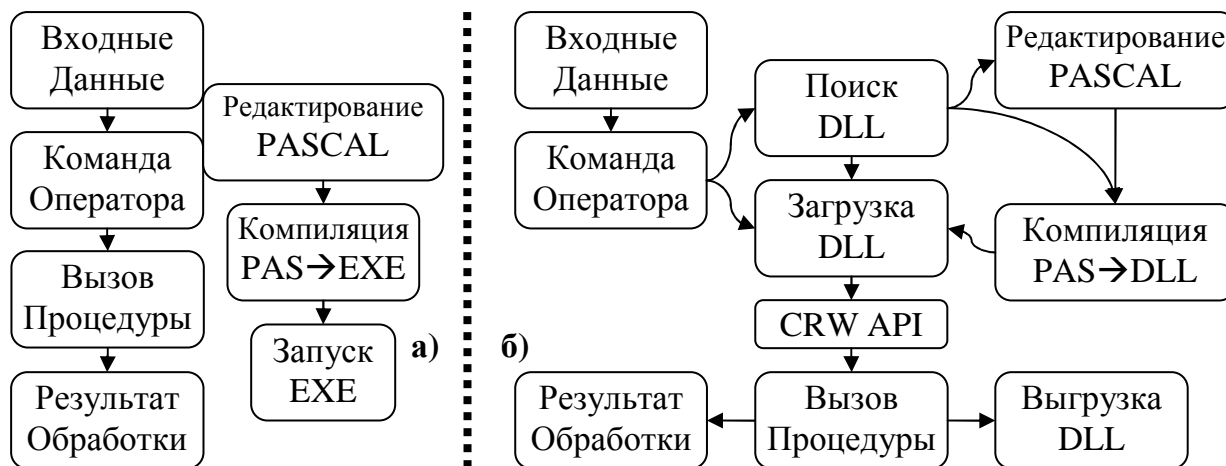


Рис.61. Сравнение обычной программы (а) и динамического расширения (б).

В технологии DPE (Рис.61,б) стадии разработки и исполнения совмещены в одном сеансе работы. Это ценное качество для исследовательских задач, где алгоритмы обработки данных часто меняются. В пакет входит встроенный компилятор и редактор проектов Object Pascal, позволяя редактировать и компилировать динамические модули DLL (Dynamic Link Library). Программный интерфейс CRW API позволяет сразу же загружать созданные DLL модули, запускать на выполнение, выгружать, редактировать, компилировать и запускать снова, не прерывая сеанса работы. Это превращает пакет в «трансформер» - программу, которая может менять свой код. Становится возможным добавлять в пакет новые функции без изменения уже отлаженного кода, что повышает его стабильность. Поэтому большинство методов offline обработки данных реализовано в виде динамических расширений DPE.

Рассмотрим реализацию динамических расширений DPE. Загружаемая DLL библиотека расширения должна экспортировать единственную функцию *CRW32_PLUGIN_FUNCTION* с одним аргументом – экземпляром класса *TCrwApi*, возвращающую ноль при успешном вызове или ненулевой код при ошибке.

Например, простейшая программа, выводящая сообщение «Hello, World!» в главную консоль пакета CRW-DAQ будет выглядеть так:

```
Library HelloWorld;  
uses ShareMem, _CrwApi;  
function PrintHelloWorld(CrwApi:TCrwApi):Integer; StdCall;  
begin  
  CrwApi.GuiApi.Echo('Hello, World!');  
  Result:=0;  
end;  
exports PrintHelloWorld name 'CRW32_PLUGIN_FUNCTION';  
end.
```

Класс *TCrwApi* объявлен в модуле *_CrwApi.Pas* (11, Листинг 1) и имеет только абстрактные виртуальные методы, конкретное наполнение которых реализовано в другом модуле, в дочернем классе внутри пакета CRW-DAQ. Класс *TCrwApi* содержит в полях другие интерфейсные классы (*SysApi*, *GuiApi*, *DaqApi*...), в свою очередь содержащие в полях ~500 функций, позволяющих обращаться к процедурам и данным пакета CRW-DAQ, которые необходимы для создания прикладных программ. Например, вызов *CrwApi.GuiApi.Echo(...)* выводит сообщение в главное консольное окно. Таким образом, через один аргумент *CrwApi* библиотека DLL получает доступ к большому набору функций и данных, содержащихся в скомпилированном и отлаженном ядре пакета CRW-DAQ. Описанный способ создания динамических расширений разработан автором самостоятельно и в литературе не встречался.

Отметим интересную особенность технологии DPE: хранимой единицей в ней является исходный программный код модулей на языке Object Pascal, который при отсутствии DLL тут же компилируется встроенным транслятором (раздел 1.4.2.4, Рис.61). Иначе говоря, технология DPE использует принцип JIT (Just In Time) компиляции «на лету», подобно технологиям Java и .NET.

5. Построение отказоустойчивых систем управления

Устройство пакета (раздел 1.4.2.6) является выражением общего принципа построения высоконадежных, отказоустойчивых систем управления (Рис.62), в основе которого лежат идеи дублирования (избыточности), параллелизма, изоляции, и иерархии управления [69]. Если взять простейшую программу сбора данных с централизованным управлением (Рис.62,а), где сбором данных занимается один процесс, то такая система не будет отказоустойчивой. Сбой в управляющей программе влечет потерю управления над всей системой.

Надежность = Дублирование + Иерархия + Изоляция + Параллелизм.

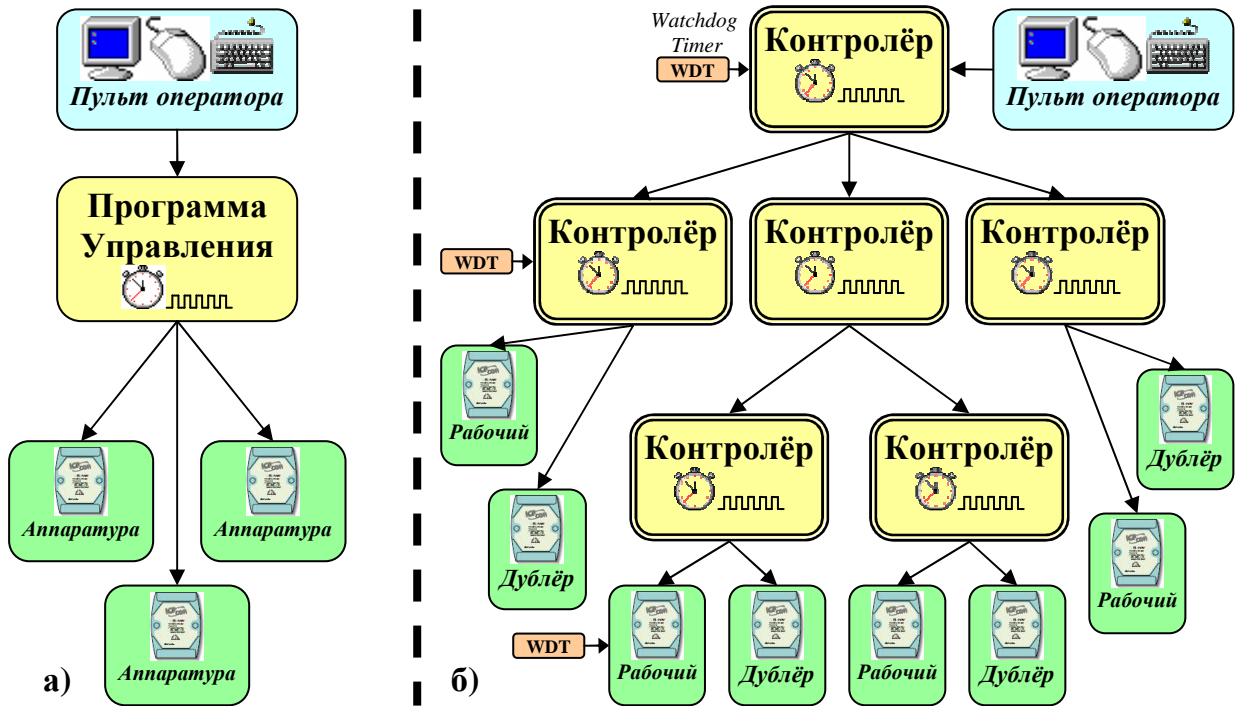


Рис.62. Сравнение обычной централизованной (а) и высоконадежной иерархичной (б) структуры системы управления.

Чтобы сделать систему отказоустойчивой, надо отказаться от централизации и ввести иерархическую систему управления, основанную на автономных, параллельно работающих и изолированных управляющих процессах - контролёрах. Каждый контролёр имеет управляемый им набор аппаратуры (Рис.62,б), а для высокой надежности и отказоустойчивости – дублирующий (избыточный) набор аппаратуры, обозначенные на рисунке как «рабочий» и «дублёр» соответственно. Контролёр следит за состоянием аппаратуры и при необходимости отключает от управления вышедший из строя рабочий набор аппаратуры, заменяя его дублёром. Однако и сам контролёр может содержать ошибки, поэтому будет логично, чтобы за ним следил другой процесс-контролёр более высокого уровня, наблюдающий за состоянием подчиненных контролёров и перезапускающий их в случае сбоя или повисания. Иерархия продолжается вверх и далее, пока не останется один контролёр верхнего уровня, управляющий системой в целом. Принципиально важно, чтобы процессы работали параллельно и были максимально изолированы друг от друга, чтобы возможная ошибка локализовалась в одном процессе, не распространяясь на другие. Для повышения надежности можно также применять сторожевые таймеры (WDT, WatchDog Timer), перезапускающие процессы, которые долго не подают признаков жизни, т.е., возможно, повисшие.

Построенная таким образом система устойчива к сбоям как аппаратным (за счет дублирования), так и программным (за счет иерархии управления и изоляции ошибок). Параллелизм, как необходимое условие, естественно возникает из требования отказоустойчивости и живучести системы, т. е. сохранения её работоспособности при выходе из строя отдельных элементов (см. 10).

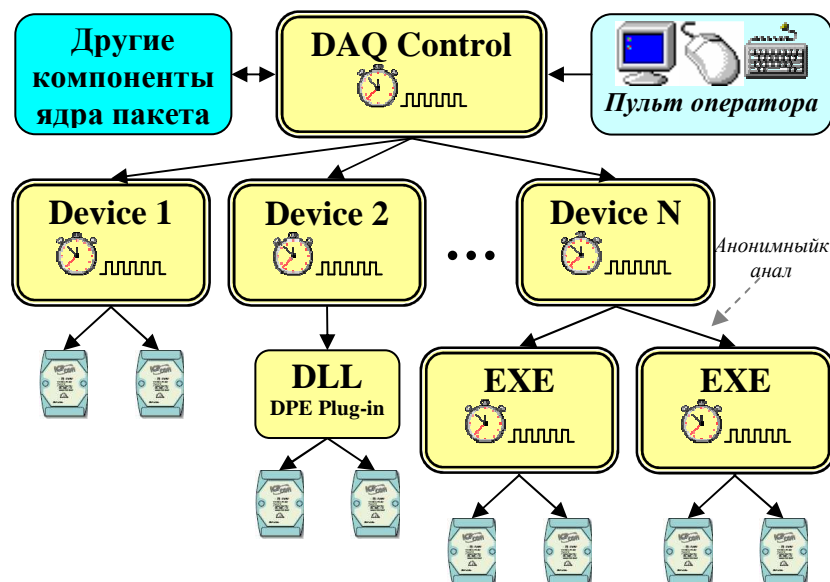


Рис.63. Иерархия управления в DAQ-системе пакета CRW-DAQ.

Если описывать структуру DAQ-системы в этих терминах, то контролёром верхнего уровня будет компонент DAQ Control (Рис.5), управляющий запуском и остановкой подчиненных ему контролёров – устройств *Device1...Device N* (Рис.63), которые входят в состав компонента DAQ Runtime и управляют аппаратурой непосредственно, либо через загружаемое расширение DPE (*DLL*). Виртуальная машина DAQ Pascal (Рис.14) обеспечивает изоляцию управляющих процессов.

Запускаемые из программ DAQ Pascal дочерние процессы (*EXE*) являются подчиненными контролёрами второго уровня. Изоляция процессов и дополнительные уровни иерархии управления повышают общую устойчивость системы. Расширения DPE не изолированы, потому их используют только для драйверов высокоскоростной аппаратуры, если скорости DAQ Pascal не хватает.

Для обеспечения работы АСКУ в реальном времени и быстрой реакции на события в системе управления, компонент DAQ разработан с учетом принципов параллелизма и разделения приоритетов, с максимальным использованием приоритетной вытесняющей многозадачности современных ОС. Параллелизм в АСКУ возникает не из желания ускорить вычисления путем разделения задачи

между процессорами, как в вычислительной математике, а потому, что реальный мир параллелен и в нем одновременно происходит много процессов. Адекватно описать это можно только в параллельных вычислительных системах.

Обычные вычислительные или интерактивные программы имеют одну очередь событий и последовательную организацию вычислений (Рис.64,а), при которой задача разбивается на ряд последовательно выполняемых процедур.

Для систем управления в реальном времени этот подход непригоден, так как в общей очереди сообщений события с низким приоритетом (обработка, отображение) и высоким (измерения, управление) обрабатываются на равных правах, а значит, время реакции АСКУ становится непредсказуемым. Поэтому все АСКУ, разрабатываемые в пакете CRW-DAQ, имеют параллельную организацию вычислений (Рис.64, б). Задача разбивается на ряд параллельно работающих служб (программных потоков или процессов), обменивающихся сообщениями и общими данными. Каждый поток обрабатывает только события одного приоритета, что позволяет высокоприоритетным потокам управления вытеснять при необходимости низкоприоритетные служебные потоки и работать без задержек.

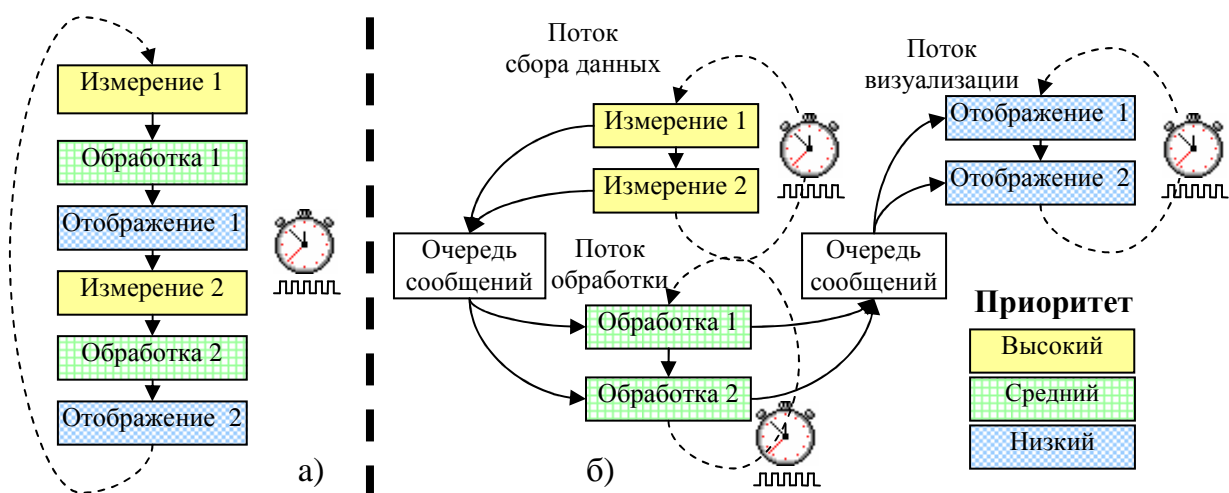


Рис.64. Последовательная (а) и параллельная (б) организация вычислений.

Сама структура DAQ-системы, построенной как многопоточное приложение, понуждает использовать параллелизм и приоритетную многозадачность, так как каждый драйвер, системная служба и прикладная программа на языке DAQ Pascal работает в своем программном потоке с регулируемым приоритетом и частотой опроса (Рис.16). Каждый из потоков решает свою простую и четко заданную задачу (обслуживает аппаратуру, обрабатывает данные) и взаимодействует с другими потоками через сообщения и хранилище RTDB. Регулируя приоритет и частоту

опроса, можно добиться снижения влияния второстепенных потоков (визуализации и архивирования) на критически важные потоки (сбора данных и управления). Многопоточный режим организуется естественным образом, не требуя больших усилий от прикладного программиста, так как все прикладные библиотеки и встроенные функции пакета созданы для работы в многопоточной среде.

Кроме многопоточного режима поддерживается также многопроцессный режим работы алгоритмов. Благодаря созданной библиотеке функций, прикладные программы на языке DAQ Pascal могут запускать другие автономные программы в виде дочерних процессов, взаимодействующих с ними по анонимному каналу (Рис.16 и Приложение 11, Листинг 3). Обычно дочерние процессы реализуют общие службы (например, сервер речевых сообщений, DIM и Web сервер) или драйверы устройств. Выделение служб и драйверов в отдельные процессы повышает отказоустойчивость системы, так как сбой дочернего процесса не приводит к остановке всей системы, а «повисшие» или «погибшие» дочерние процессы могут быть легко перезапущены прикладной программой АСКУ.

При использовании многоядерного процессора каждый поток (т.е. прикладную программу DAQ Pascal) или дочерний процесс можно привязать к желаемому процессору и добиться повышения стабильности частоты опроса критически важных потоков (процессов), что важно для АСКУ реального времени.

Для понимания принципов построения DAQ-системы в пакете CRW-DAQ полезна модель, представляющая работу АСКУ как задачу навигации (выбора допустимой траектории) в фазовом пространстве переменных (Приложение 6).

6. Работа АСКУ как навигация в фазовом пространстве

Полезной моделью для понимания принципов построения DAQ-системы в пакете CRW-DAQ является представление о работе системы управления АСКУ как о задаче навигации (выбора допустимой траектории) в фазовом пространстве переменных для достижения некоторой целевой области в нем (Рис.65).

В классической вычислительной программе (1) имеется определенный набор исходных данных и некоторая четкая цель. Задача программы состоит в том, чтобы путем выполнения ряда процедур достигнуть её и получить результат вычислений, который жестко детерминирован входными данными и алгоритмом, то есть при одинаковых исходных данных и алгоритме результат вычислений повторяется. Для

описания подобных программ часто применяется язык блок-схем, изображающих алгоритм вычисления искомого результата в виде ряда взаимосвязанных процедур.

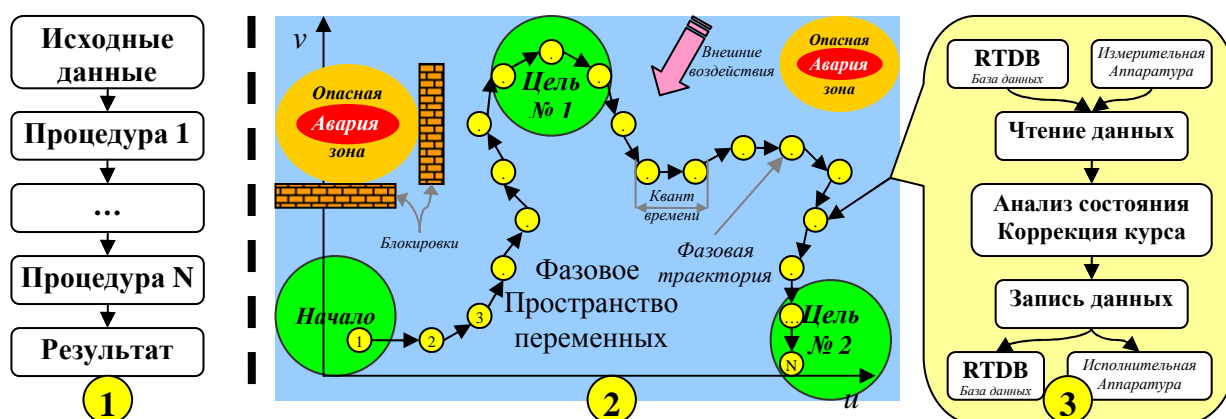


Рис.65. Модель классической программы (1) и программы управления АСКУ (2).

Ситуация резко изменится, как только мы добавим в систему измерительные и исполнительные устройства, связанные с реальным физическим объектом. Прежде всего, система потеряет жестко детерминированный характер, по крайней мере, с точки зрения программы, поскольку измерительные устройства и сам объект измерения подвержены неизвестным внешним воздействиям, зачастую случайным и к тому же непредсказуемо меняющимся во времени. Это значит, что нельзя ожидать точной повторяемости работы алгоритмов, даже если исходные данные и алгоритм неизменны. Кроме того, управляемый объект имеет внутреннюю динамику и его нельзя заставить принять произвольное состояние путем программного воздействия. Блок-схемы становятся неадекватным средством описания задачи, ведь зная состояние объекта в некий момент времени, нельзя гарантировать его сохранение в следующий момент даже при отсутствии программного воздействия. Другой проблемой при создании систем управления является то, что зачастую как начальные условия, так и цели работы АСКУ не могут быть четко заданы, а известны лишь приближенно, в виде ряда желательных условий. Кроме того, исследовательские АСКУ обычно имеют не одну, а множество целей и режимов работы.

По указанным причинам более подходящей моделью для систем управления будет следующая (2). Пусть имеется система, однозначно описываемая набором переменных состояния v (от измерительной аппаратуры), параметров управляющего воздействия u (для исполнительной аппаратуры) и внутренних переменных x , нужных алгоритму управления. Совокупность $\varphi_{(t)} : \{v_{(t)}, u_{(t)}, x_{(t)}\}$

зависящих от времени t переменных, однозначно определяющих состояние АСКУ, образует многомерное фазовое пространство. Состояние АСКУ в каждый момент времени описывается точкой в этом пространстве, а её динамика представляется некоторой траекторией в фазовом пространстве. Внешние воздействия описываются в виде «силового поля», воздействующего на фазовую траекторию. Начальное состояние, всегда известное с некоторой погрешностью, а также цели работы АСКУ можно представить как некоторые области фазового пространства, через которые должна проходить желаемая фазовая траектория. Ряд областей фазового пространства может квалифицироваться как «опасная зона» или даже «авария» и их следует избегать. Задачу системы управления можно теперь сформулировать как проблему безопасной навигации в фазовом пространстве, то есть построения фазовой траектории, проходящей через желаемые целевые зоны фазового пространства, но не попадающей в опасные зоны, несмотря на внешние воздействия. Имеющиеся в системах управления блокировки можно рассматривать как охранительные конструкции, предотвращающие приближение фазовой траектории к аварийным зонам (Рис.65).

Реальное время непрерывно, но компьютерное время квантовано, поэтому фазовую траекторию следует представить дискретной последовательностью точек в фазовом пространстве $\varphi_i : \{v_i, u_i, x_i\}$, соответствующих очередному кванту времени t_i системы управления. Хотя из-за внешних воздействий и внутренней динамики объекта управления нельзя считать детерминированным состояние системы в целом, но можно положить детерминированным (фиксированным) её состояние в течение хотя бы отдельного кванта времени. Тогда алгоритм управления в каждом кванте времени становится возможным описать в привычных терминах процедурного программирования и блок-схем (3). При этом состояние системы между квантами времени по-прежнему считается недетерминированным. Можно сравнить такой подход со стробоскопом, позволяющим видеть движущийся объект как последовательный ряд отдельных статических фотоснимков.

Получив управление в каждом новом кванте времени, первым делом надо определить, в какой точке фазового пространства оказалась система с момента предыдущего кванта времени под воздействием внешних факторов и внутренней динамики управляемого объекта. Для этого считываются каналы измерительной

аппаратуры и теги из хранилища данных RTDB. На основании прочитанных данных алгоритм управления (прикладная программа) анализирует состояние и классифицирует область фазового пространства, в которой оказалась система. Затем на основании этого анализа принимается решение, какие управляющие воздействия необходимы для коррекции курса, чтобы в следующем кванте времени траектория системы приблизилась к целевой области или удалилась от опасной зоны фазового пространства. Наконец, управляющие воздействия записываются в регистры исполнительных устройств, а переменные состояния в хранилище, после чего работа завершается до следующего кванта времени.

Наиболее естественной формой задания вышеупомянутых зон фазового пространства (начальной, целевой, аварийной) является набор неравенств, ограничивающих их. Например, задачу стабилизации температуры T можно рассматривать как задачу построения фазовой траектории, попадающей в целевую зону, заданную неравенством $|T - T_0| < \Delta T$, где T_0 - целевая температура, ΔT - допуск, при этом неравенство $T > T_{\max}$ задает опасную зону, которой следует избегать. Неудивительно поэтому, что управляющие алгоритмы в данной модели строятся не как последовательный «ряд процедур», а как «набор правил», описывающих «коррекцию курса» при обнаружении определенных условий. Так, алгоритм стабилизации температуры можно сформулировать в виде набора правил: «если $T < T_0 - \Delta T$, то увеличить нагрев», «если $T > T_0 + \Delta T$, то уменьшить нагрев», «если $T > T_{\max}$, то выключить нагрев и включить аварийную сигнализацию».

Вышеописанная модель построения алгоритмов управления АСКУ как «набора правил» вместо «ряда процедур» названа автором «асинхронным программированием». Название выбрано из-за принципиального отказа в алгоритмах управления от синхронизирующих задержек или циклов ожидания событий синхронизации. Вместо этого алгоритм управления формулируется как «набор правил навигации в фазовом пространстве» (Рис.65). Вся DAQ-система в пакете CRW-DAQ реализована в рамках этой модели, а её структура буквально понуждает прикладного программиста использовать асинхронный стиль программирования. В то же время основной инструмент создания прикладных программ (DAQ Pascal) идеально подходит именно для этого стиля (раздел 1.4.2.5).

Принцип «асинхронного» программирования, как всякий иной, имеет недостатки и достоинства. Недостатком может показаться, например, его избыточность и некоторое снижение производительности, ведь в каждом кванте времени прикладная программа должна заново считывать данные, анализировать состояние, вырабатывать реакцию и записывать данные. Достоинством является повышение безопасности алгоритмов управления, поскольку модель не делает (неявных) предположений о поведении объекта и потенциально всегда готова к любому событию, например, к внезапному возникновению аварийной ситуации.

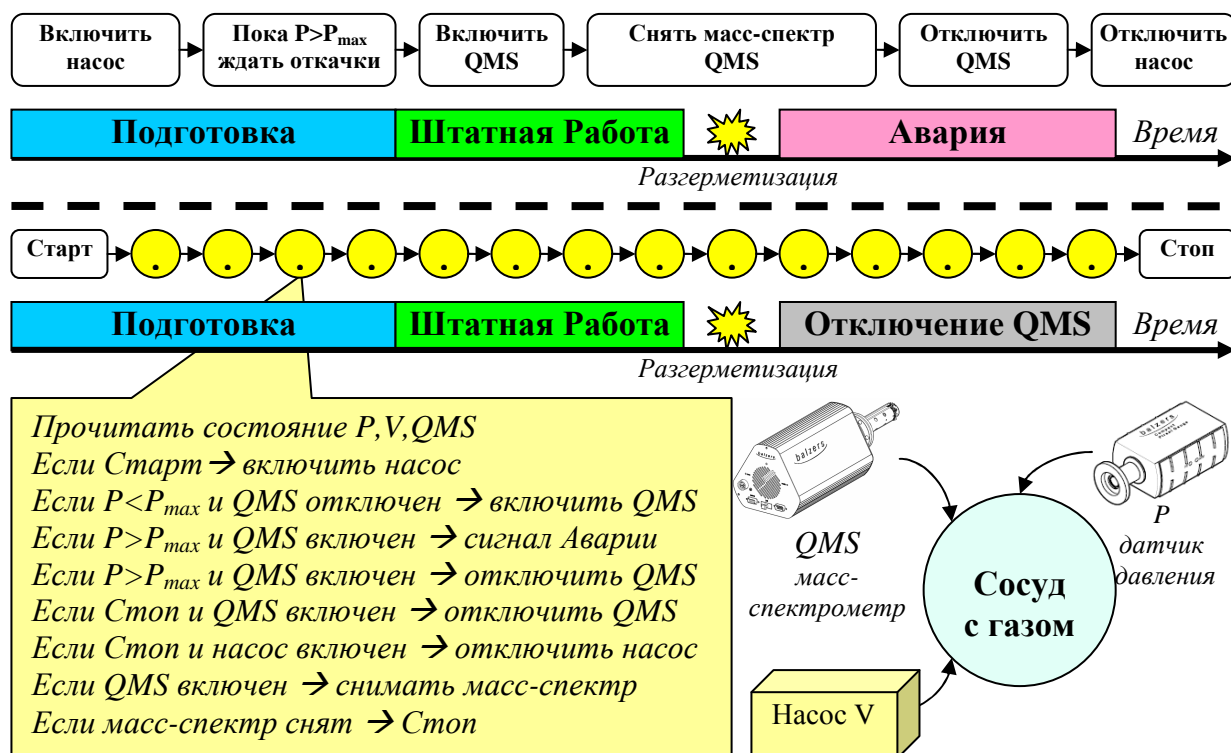


Рис.66. Сравнение безопасности процедурной и асинхронной программы.

Рассмотрим, например, простую систему газового анализатора (Рис.66) из сосуда с газом, вакуумного насоса V, датчика давления P и квадрупольного масс-спектрометра QMS. По условиям эксплуатации QMS не может работать без вакуума и при $P > P_{max}$ он выходит из строя, что рассматривается как авария. Процедурный алгоритм будет примерно такой: включить насос, откачивать вакуум пока не станет $P < P_{max}$, включить QMS, снять масс-спектр, отключить QMS, отключить насос. Однако если при снятии спектра внезапно произошла разгерметизация, то возникнет авария, ведь из факта наличия вакуума в начале измерения не следует его дальнейшее сохранение. В то же время асинхронная программа, проверяющая давление вакуума в каждом кванте времени, не допустит

аварии даже при внезапной разгерметизации. Для задач управления повышение безопасности важнее производительности, поэтому применение вышеописанного асинхронного стиля программирования при разработке АСКУ вполне оправдано.

7. Распределенные АСКУ на базе DIM и WEB технологий

Рассмотрим подробнее создание распределенных многомашинных сетевых систем управления (раздел 1.4.2.7). Хотя в пакете CRW-DAQ создано несколько сетевых библиотек, включая поддержку анонимных и именованных каналов, сокетов TCP/IP, протокола HTTP и Web сервер, но основным средством межмашинного взаимодействия в пакете выбрана технология DIM [70].

По своей функциональности и целевому назначению технология DIM аналогична широко применяемой в промышленной автоматизации технологии OPC (OLE for Process Communication, см. Web сайт www.opcfoundation.com). Обе они решают задачи распределенного сбора данных, и управления в реальном времени и используют асинхронный механизм обмена данными. Достоинствами OPC являются его стандартизация и поддержка крупными производителями оборудования, недостатками – закрытость, высокая сложность, громоздкость, относительно низкая скорость и надежность, а также то, что OPC технология поддерживается только под Windows. Достоинством технологии DIM является исключительно высокая скорость обмена данными, ограниченная только скоростью сети и достигающая 200 мегабайт в секунду при обмене данными на одном компьютере. Её отличает также высокая эффективность и надежность, простота освоения и лаконичность кода, поддержка большого числа операционных систем (Windows, Linux, QNX) и языков программирования (C, C++, Java, Fortran, Pascal), наличие исходных кодов на языке C и отсутствие лицензионных проблем. Это позволяет использовать её в любых проектах, привлекая программистов самой разной квалификации. По указанным причинам технология DIM была выбрана основной технологией сетевого обмена и интегрирована в пакет CRW-DAQ [19].

Технология DIM предоставляет «прозрачный» с точки зрения сети способ взаимодействия процессов, которые могут выполняться на одной машине или на разных машинах в сети Ethernet и совместно решать одну задачу управления. В качестве протокола нижнего уровня она использует TCP/IP, однако предлагает

конструкции более высокого уровня, позволяя абстрагироваться от деталей сетевого обмена, сосредоточив усилия на самих данных.

Технология DIM основана на классической модели клиент-сервер, но имеет ряд существенных расширений, ориентированных на распределенные системы управления реального времени. Основным понятием DIM является «сервис» - поименованный набор двоичных данных произвольной структуры. Процесс, который владеет этими данными (система сбора данных и управления), является DIM сервером, причем каждый сервис принадлежит одному серверу. Потребители данных сервиса (удаленные клиентские терминалы) - это DIM клиенты, число которых не ограничено.

В отличие от классической синхронной модели взаимодействия, когда клиент посылает серверу запрос и ждет от него ответа, технология DIM использует асинхронную модель взаимодействия. Она состоит в том, что клиенты «подписываются» на интересующий сервис и продолжают свою работу, не ожидая ответа сервера, а сервер по таймеру или при изменении данных сервиса делает «рассылку» данных сервиса всем подписанным у него на этот сервис клиентам (Рис.67). Асинхронная модель позволяет избежать циклов ожидания, что очень важно для задач управления. Клиенты также могут асинхронно посылать серверу команды, позволяющие управлять его состоянием.

Интересно подробнее сравнить подход автора к реализации распределенных систем управления в пакете CRW-DAQ на основе DIM с другими распределенными системами управления экспериментом, например, с ИВК Т-10 [80], а также MIDAS (см. <http://midas.psi.ch>). Эти системы также имеют структуру клиент-сервер, однако способ взаимодействия клиента и сервера существенно различается.

Есть две диаметрально противоположные парадигмы организации систем клиент-сервер (Рис.67). Первая парадигма (Рис.67,а), именуемая часто RPC (Remote Procedure Call) предполагает, что сервер - владелец информационных ресурсов, обслуживает запросы клиентов, являющихся потребителями этих ресурсов, которые берут у сервера интересующие данные и отсылают ему результаты обработки, если это нужно.

Взаимодействие происходит по принципу «Запрос-Ожидание-Ответ» (Request-Waiting-Reply), причем активной стороной является клиент, посылающий

запрос. Большинство систем управления базами данных (СУБД) и Internet-технологий основано на этой парадигме (например, SQL и HTTP).

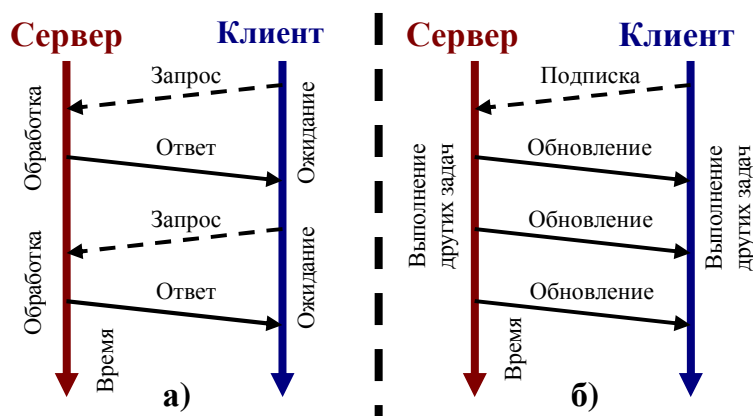


Рис.67. Сравнение двух парадигм взаимодействия систем клиент-сервер: а) запрос-ожидание-ответ (HTTP, SQL...), б) подписка-обновление (DIM, OPC...).

Вторая парадигма (Рис.67,б), реализованная, например, в технологиях OPC и DIM, основана на принципе «Подписка-Обновление» (Subscribe-Refresh). При этом клиент однократно (при подключении к серверу) делает подписку (Subscribe) на интересующий информационный сервис (Service), которым владеет сервер (например, на данные измерительной системы), и продолжает решение клиентских задач без ожидания ответа от сервера. При поступлении новых данных сервер автоматически делает рассылки (Refresh) всем клиентам, которые подписались на обновляемые информационные ресурсы. Активной стороной при этом является сервер, а обновление данных на стороне клиента имеет принудительный и асинхронный характер, обычно реализуемый через обратный процедурный вызов (Callback). Использование принципа Subscribe-Refresh характерно для систем управления реального времени.

Сравнивая две парадигмы взаимодействия клиент-сервер, надо в первую очередь понимать, что они ставят разные цели, и поэтому каждая из них оптимальна - для своего круга задач. Принцип RPC минимизирует трафик между клиентом и сервером. Он оптимален для большинства СУБД, когда клиента интересует только малая часть из большого серверного хранилища данных. При этом клиент посылает запрос, описывая требуемый информационный ресурс (например, URL в случае HTTP или SQL запрос в случае СУБД). Сервер делает выборку и отправляет клиенту только то, что он запросил. Если система АСКУ ориентирована на сбор и последующий offline анализ большого объема данных, как

системы ИБК Т-10 или MIDAS, то принцип RPC уместен и основное внимание занимают механизмы хранения и удобного доступа к данным. В то же время в RPC всегда присутствует неопределенное по времени ожидание ответа на запрос. Кроме того, активной стороной RPC является клиент, поэтому если клиент по каким-то причинам не отправляет серверу запрос, он не будет вовремя уведомлен о состоянии объекта, даже если с ним случилась авария. Это недопустимо для систем управления, где оповещение оператора о возникшей радиационной или иной опасности должно быть незамедлительным и принудительным.

Принцип Subscribe-Refresh ставит целью не минимальный трафик, а минимальное время отклика системы на события в объекте управления. Если клиент подписан на информационный ресурс, он немедленно и принудительно получит его обновление, даже если в данный момент это не представляет для него интерес, поэтому трафик заведомо не будет оптимальным. Однако в данной парадигме отсутствует ожидание, как на стороне сервера, так и на стороне клиентов, которые оповещаются о событиях в системе по инициативе сервера немедленно и принудительно. Для систем управления потенциально опасными объектами в реальном времени требования к своевременному оповещению и времени реакции АСКУ важнее минимизации сетевого трафика, обычно небольшого. Поэтому принцип «Подписка-Обновление» взят автором за основу сетевых средств пакета CRW-DAQ и лежит в основе большинства других систем SCADA (Supervisory Control and Data Acquisition).

Другим важнейшим понятием DIM является сервер имен DNS (Distributed Name Server). В классической модели клиент, чтобы подключиться к серверу, должен точно знать его сетевые координаты (имя или IP адрес сервера, номер порта). В сети из десятков машин, когда каждый из десятков клиентов использует данные десятков серверов, это представляется весьма сложной задачей, тем более что система является динамической: машины в сети появляются и исчезают, меняются их адреса и т.д. В технологии DIM клиент и сервер должны знать только имя DNS сервера и имя сервиса, не зависящее от сетевой структуры системы. Сервер имен хранит динамическую базу данных клиентов и серверов - владельцев сервисов, так что клиент, подписываясь на сервис, узнает координаты сервера через сервер имен. Таким образом, зная имя DNS сервера и имя сервиса, можно

строить систему любого уровня сложности, не заботясь более об именах и адресах серверов и клиентов. Более того, наличие DNS позволяет делать “горячую” замену серверов (например, при выходе одного сервера из строя заменять его другим), не нарушая работу клиентов, что в классической модели клиент-сервер невозможно.

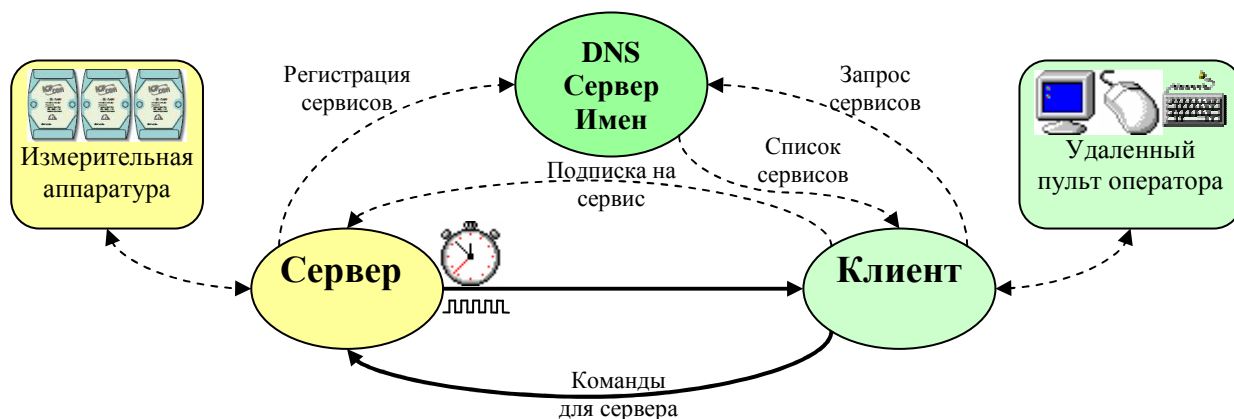


Рис.68. Диаграмма потока данных технологии DIM.

Каждый DIM сервер, начиная работу, регистрирует свои сервисы в базе данных сервера имен (Рис.68). Клиент, желая подписаться на интересующий сервис, посылает запрос серверу имен и получает от него список сервисов с данными о сервере, который обслуживает этот сервис. Получив адрес сервера, клиент устанавливает с ним соединение и подписывается на интересующий сервис. После этого сервер по таймеру или при обновлении данных сервиса пересылает обновленные данные подписанным клиентам, а клиент при необходимости посылает серверу команды для управления его состоянием.

Хотя сама технология DIM взята за основу сетевых средств «как есть», её интеграция в пакет CRW-DAQ не была тривиальной (Рис.69). Дело в том, что технология DIM основана на загрузке динамической библиотеки (DIM.DLL) и обратных вызовах пользовательских процедур (Callback), выполняемых в контексте управляющего процесса, что является потенциально опасным с точки зрения его стабильности.

Для сохранения высокой надежности работы пакета было решено вынести поддержку DIM в отдельный процесс (DIMSRV.EXE), обменивающийся с пакетом CRW-DAQ по анонимному каналу связи (pipe) при помощи работающего «на общих основаниях» драйвера &DIMSRV, написанного на языке DAQ Pascal (Рис.16). Драйвер передает и принимает по каналу данные и взаимодействует с ядром пакета или другими прикладными программами посредством общих тегов из

хранилища данных RTDB и обмена сообщениями. Неизбежное при этом снижение скорости обмена стало ценой за сохранение высокой надежности работы пакета.

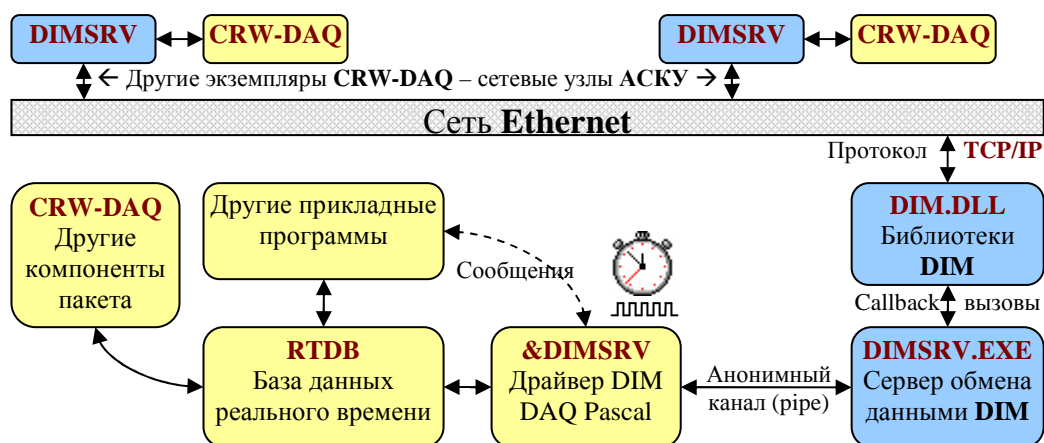


Рис.69. Принцип интеграции технологии DIM в пакет CRW-DAQ.

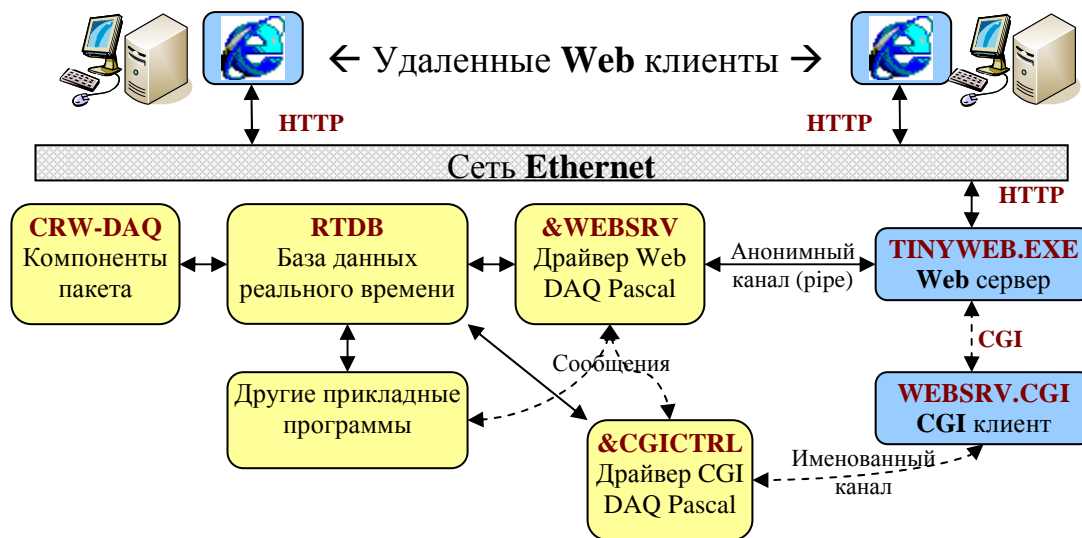


Рис.70. Принцип интеграции технологии Web в пакет CRW-DAQ.

Широкое распространение сети Internet привело к популярности Web-технологий. Поэтому второй сетевой технологией, интегрированной в пакет, стала поддержка Web протокола HTTP. За основу было решено взять свободно распространяемый в исходных кодах Web-сервер TinyWeb, доступный на сайте www.ritlabs.com/tinyweb/, дополнив его необходимыми компонентами, обеспечивающими его запуск, завершение и обработку CGI запросов. В результате АСКУ под управлением CRW-DAQ может работать как Web сервер, а удаленные Web клиенты могут наблюдать данные и управлять установкой, пользуясь обычным обозревателем Internet Explorer. Для создания Web-приложения прикладной программист должен сконфигурировать встроенный в пакет Web-сервер и написать на языке DAQ Pascal несложную программу, формирующую

динамическую страницу на языке HTML, которая и передается клиенту по его запросу. Важно, что при этом от программиста требуется самый минимальный уровень подготовки в области Web и HTML.

Интеграция Web в пакет CRW-DAQ реализована по следующему принципу (Рис.70). Удаленный клиент открывает Web обозреватель и посылает HTTP запрос с адресом сервера. Встроенный Web сервер TYNWEB.EXE, запускаемый и управляемый по анонимному каналу прикладной программой &WEBSRV, получает запрос и запускает специальную утилиту WEBSRV.CGI, передавая ей запрос по протоколу CGI. Эта утилита связывается по именованному каналу с прикладной программой &CGICTRL и передает ей свой CGI запрос. Программа в ответ формирует HTML страницу, взяв нужные данные из хранилища RTDB, и передает ее по цепочке &CGICTRL → WEBSRV.CGI → TYNWEB.EXE обратно Web серверу, который пересылает ее клиенту. Описанная схема обработки запросов HTTP может показаться слишком сложной, но это было оправданно желанием автора использовать стандартные инструменты и протоколы (в данном случае это готовый Web сервер TinyWeb и протокол CGI). Важно отметить, что нужные программы (&WEBSRV, WEBSRV.CGI) уже созданы автором и никогда не меняются, так что для разработки Web сервера от прикладного программиста требуется только написание по имеющемуся шаблону несложной прикладной программы &CGICTRL на языке DAQ Pascal, чтобы сформировать конкретную HTML страницу для передачи клиенту. Это делает задачу создания Web сервера доступной даже для программистов начального уровня.

Польза от использования Web-технологий для АСКУ состоит в том, что широкое распространение Web снимает проблему создания клиентского программного обеспечения, ведь Web обозреватель доступен на любом компьютере. Это позволяет снизить затраты на разработку прикладных программ для установок, так как заботиться приходится только о программном обеспечении для сервера. Правда, возможности Web ограничены двумя факторами: высокими накладными расходами при передаче данных по HTTP и дисциплиной его работы (инициировать обновление данных может только клиент). Поэтому использование Web оправдано только тогда, когда объем передаваемых данных сравнительно невелик, а быстрой реакции на события со стороны клиента не требуется.

8. Конфигурирование измерительных систем CRW-DAQ

Программное обеспечение АСКУ (раздел 1.4.2.6) располагается в отдельном каталоге, содержащем обычно следующие подкаталоги: Bitmaps (файлы **.bmp* изображений мнемосхем), Calibr (файлы **.cal* калибровок), Circuits (файлы **.crc* описаний мнемосхем), Config (файлы **.cfg* конфигурации АСКУ), DaqPas (файлы **.pas* прикладных программ DAQ Pascal), Data (файлы **.dat* измеренных данных), Help (файлы **.htm* и **.doc* справочной документации), Sound (файлы **.wav* звуковых сообщений). Один из конфигурационных файлов является главным, является корнем дерева ссылок на другие конфигурационные файлы, программы, изображения и документы, необходимые для работы АСКУ. Конфигурационные файлы содержат следующие обязательные секции, с которых начинается интерпретация АСКУ: *[ConfigFileList]*, *[DAQ]*, *[TagList]*, *[DataStorage]*, *[DeviceList]*, *[Windows]*.

Секция *[ConfigFileList]* содержит список *ConfigFile* ссылок на файлы, которые включаются в конфигурацию. Интерпретатор добавляет текст этих файлов в конец текста разбираемой конфигурации, как будто все эти файлы образуют единый текст. Каждый из файлов может содержать свою секцию *[ConfigFileList]*, так что получается дерево файлов, которое суммируется в один текст. Например:

```
[ConfigFileList]
ConfigFile = One.cfg, Two.cfg
ConfigFile = ..\Config\Three.cfg, ..\Help\Four.cfg
```

Секция *[DAQ]* описывает общие параметры DAQ-системы, такие как единицы измерения времени *TimeUnit*, ссылку на файлы справки *HelpFile* и каталог данных *DataPath* и т.д. Например:

```
[DAQ]
TimeUnit = 60 ; единицы измерения времени, секунд
DataPath = ..\data ; каталог для хранения данных
HelpFile = ..\help\demo.htm ; файл справки
```

Секция *[TagList]* содержит список деклараций тегов RTDB, т. е. общих для всех прикладных программ поименованных скалярных переменных. При описании тега указывается его тип (*integer*, *real* или *string*) и начальное значение. Например:

```
[TagList]
Demo.Btn = integer 1 ; целочисленный тег
Demo.Win = string DEMO.CTRL ; строковый тег
Demo.Pow = real 50.0 ; вещественный тег
```

Секция *[DataStorage]* содержит список деклараций кривых *Curve* для хранилища данных реального времени RTDB, то есть общих для всех прикладных

программ и драйверов поименованных динамических массивов $\{(x_i, y_i), i = 1 \dots N\}$ для хранения зависимостей $y_{(x)}$. Как правило, x - это время в единицах *TimeUnit*, а y - измеряемая величина. При декларации кривой указывается также ряд её атрибутов: начальная длина, шаг буферизации, цвет и стиль точек и линий. Например:

```
[DataStorage]
ADC.I = Curve 0 100 Black 0 2 ; кривая тока, стиль точек 0, линий - 2
ADC.U = Curve 0 100 Blue 1 1 ; кривая напряжения, стиль точек 1, линий - 1
```

Секция *[DeviceList]* содержит список деклараций устройств *Device* для описания используемых драйверов или прикладных программ (Рис.16). Каждая декларация определяет имя, класс (*software, adam, pcl*) и зависящий от класса тип устройства. Класс *software* включает устройства, связанные с прикладными программами: *program* (программа на языке DAQ Pascal), *script* (программа на языке DAQ Script), *dialog* (устройство для организации диалогового интерфейса пользователя). Класс *adam* включает устройства, связанные с драйверами устройств на шинах RS-232, RS-485: модулей серии ADAM-4000, модулей серий I-7000 и I-87000, *adam_slot, rsdaqhost* (устройств для связи в сети RS-485), *balzers-tpg256, balzers-tpg252* (контроллеров датчиков вакуума фирмы Balzers или Pfeiffer). Класс *pcl* содержит драйверы устройств на шине ISA, такие как *DIO-24, DIO-48, DIO-144, PCL-731* и т.д. После декларации устройства в одноименной с ним секции располагается описание параметров, зависящее от его класса и типа. Например, все устройства содержат комментарий *Comment* и период опроса *InquiryPeriod*, а устройства класса *software* содержат вдобавок частоту опроса и приоритет программного потока *DevicePolling*. Устройства типа *program* содержат ссылку *ProgramSource* на файл с текстом программы на языке DAQ Pascal, а устройства класса *adam* содержат номер *port* COM порта и сетевой адрес модуля *address*. Кроме того, все устройства имеют аналоговые и цифровые входы и выходы, к которым можно подключать кривые из секции *[DataStorage]* при помощи конструкции «*Link ... with curve ...*». Например:

```
[DeviceList]
&Demo.DRV = device software program ; объявление устройства
[&Demo.DRV] ; описание устройства:
Comment = DAQ Pascal program ; комментарий
InquiryPeriod = 1 ; период опроса миллисекунд
DevicePolling = 1, tpTimeCritical ; период опроса и приоритет потока
ProgramSource = ..\DaqPas\DemoDRV.pas ; ссылка на файл с кодом программы
AnalogInputs = 2 ; число аналоговых входов
AnalogOutputs = 2 ; число аналоговых выходов
Link AnalogInput 0 with curve ADC.I ; аналоговые входы
Link AnalogInput 1 with curve ADC.U
Link AnalogOutput 0 with curve U.VOLT ; аналоговые выходы
```

```

Link AnalogOutput 1 with curve I.AMPER
[DeviceList]
&I7018 = device adam ADAM-7018
[&I7018]
Comment      = Драйвер для I7018, адрес $01
InquiryPeriod = 10 ; Период опроса, ms
Port         = 1   ; Номер COM-порта
Address      = $01 ; Адрес устройства в сети RS-485
UsesChecksum = 1   ; Использовать контрольные суммы?
RangeCode   = $01 ; $00=15mV, $01=50mV, $02=100mV, etc.
DataFormat  = 2   ; 0=Engineering,1=%FSR,2=Hex,3=Ohms
Link AnalogOutput 0 with curve ADC.I history 200
Link AnalogOutput 1 with curve ADC.U history 200

```

Секция *[Windows]* содержит список деклараций окон *Curve_Window*, *Tab_Window*, *Circuit_Window* для описания используемых окон кривых, таблиц и мнемосхем соответственно (раздел 1.4.2). Каждая декларация определяет имя и тип окна. После декларации окна в одноименной с ним секции должно быть расположено описание параметров, содержание которого определяется его типом. Например, окна кривых содержат список *CurveList* отображаемых в окне кривых, а окна мнемосхем содержат ссылку *Circuit* на файл описания мнемосхемы и конструкции «*Link ... with ...*» для установления связи сенсоров (визуальных элементов мнемосхем) с тегами и кривыми RTDB. Например:

```

[Windows]
GraphU = Curve_Window
[GraphU]
CurveList = ADC.U, U.VOLT
[Windows]
AHIT.CTRL = Circuit_Window
[AHIT.CTRL]
Circuit = ..\CIRCUITS\AHIT_CTRL.CRC
Link sensor I with tag ADC.I device &Demo.DRV
Link sensor U with tag ADC.U device &Demo.DRV

```

Подробное описание форматов конфигурационных файлов, калибровок, мнемосхем содержится в электронной документации пакета.

9. Система ограничения прав доступа к АСКУ

Работа тритиевых установок связана с потенциальной радиационной опасностью, поэтому их системы управления должны включать систему прав доступа, исключающую несанкционированные действия посторонних лиц.

Реализованная в пакете CRW-DAQ система прав доступа и защиты опирается на систему защиты Windows. Рассмотрим сначала защиту на уровне пользовательского интерфейса (Рис.71). Операционная система конфигурируется так, что графическая оболочка Windows работает с правами ограниченного пользователя, а измерительный пакет CRW-DAQ запускается с правами администратора, т.к. ему нужен доступ к аппаратуре. Для запуска программ с

правами администратора (но без ввода его пароля) из-под пользователя с ограниченными правами специально разработана программа AdmiLink, доступная на Web сайте www.crw-daq.ru/download/doc/admilink/, использующая зашифрованную учетную запись администратора. Файловая система гарантирует невозможность прямого изменения конфигурационных файлов пользователем, так как они созданы под администратором и защищены системой безопасности Windows. Конфигурацию можно изменить из среды пакета CRW-DAQ, но там существует своя защита, в которой предусмотрено 4 уровня доступа:

- Lock — компьютер заблокирован, запрещены все операции, кроме смены уровня доступа. Применяется для временной блокировки компьютера.
- Guest — уровень гостя (неопытного пользователя), защищен паролем. Позволяет строить графики, наблюдать состояние параметров, но доступ к управлению измерительной системой закрыт.
- User — уровень (опытного) пользователя, защищен паролем. Позволяет управлять измерительной системой (запуск, остановка, завершение работы), однако редактирование прикладных программ или конфигураций запрещено. Это гарантирует целостность конфигурации и алгоритмов измерительной системы, предохраняет ее от несанкционированных изменений.
- Root — уровень администратора, защищен паролем. Администратору разрешены все операции, включая редактирование и компиляцию прикладных программ. Этот уровень предназначен только для разработчиков и администраторов измерительных систем.

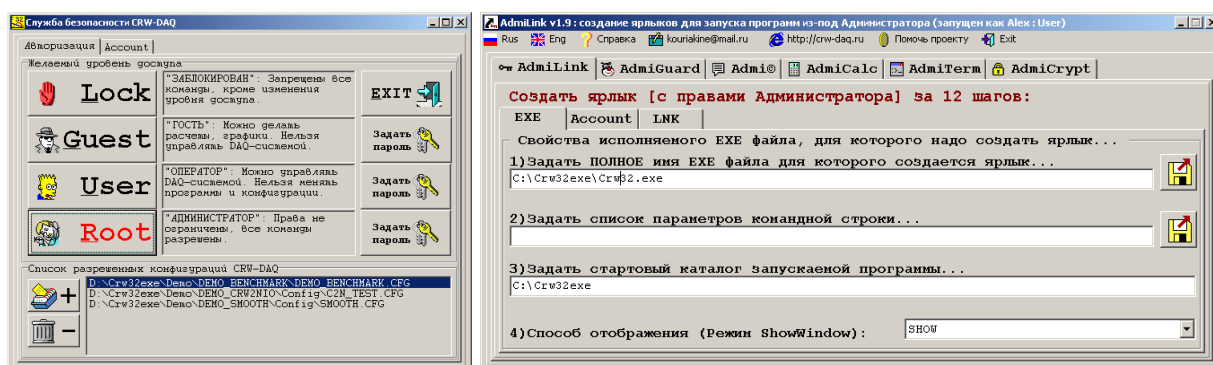


Рис.71. Внешний вид системы защиты пакета CRW-DAQ и программы AdmiLink.

Обычно, если для данной установки требуется система защиты, вся рутинная работа происходит на уровне пользователя или гостя. Для выполнения критических

действий оператор вводит пароль для временного повышения уровня доступа, а по их завершении возвращается на пониженный уровень доступа. Понижение уровня доступа выполняется одной кнопкой (F11) и не требует пароля, что позволяет быстро заблокировать компьютер.

При работе в сети кроме защиты на уровне интерфейса пользователя необходима также сетевая защита. Самым уязвимым звеном при этом является сервер, принимающий от клиентов команды - сервисы в случае DIM и HTTP запросы в случае Web. В обоих случаях DIM или HTTP команды передаются по открытому каналу связи и в принципе могут быть посланы любым пользователем. Поэтому для защиты сервера несанкционированного доступа по сети используется шифрование. В пакет включена система шифрования, использующая шесть свободно опубликованных алгоритмов: (Blowfish, Gost, RC2, RC4, RC5, RC6) [81]. В алгоритме шифрования учитываются такие параметры как уровень доступа, имя и пароль пользователя, имя удаленного компьютера, его IP и MAC адрес. Сервер (DIM или Web) хранит таблицу допустимых параметров пользователей. При получении от клиента зашифрованной команды сервер пытается расшифровать её, но это удастся сделать только в том случае, если пользователь зашифровал её, используя те же самые параметры, что указаны в таблице у сервера. Это гарантирует аутентичность получаемых сервером команд. В то же время трафик от сервера к клиентам в пакете CRW-DAQ открыт и не шифруется.

10. Особенности программной реализации пакета CRW-DAQ

Работа на тритиевых установках носит длительный и непрерывный (несколько месяцев) характер, при этом радиационный контроль и аварийная блокировка должна работать исключительно надежно. Поэтому при создании программного обеспечения АСКУ, как системного (пакет CRW-DAQ), так и прикладного (алгоритмы управления), высший приоритет имела его безопасность и долговременная надежность. Ради повышения надежности иногда приходилось жертвовать некоторым снижением скорости работы или усложнением программ, однако предпочтение отдавалось вопросам безопасности АСКУ. Надежность программ имеет два аспекта – рассмотренный выше структурный (раздел 1.4.2.6, Рис.62) и технологический, связанный с методами безопасного программирования.

Технологии безопасного программирования - одна из самых актуальных тем развития компьютерной индустрии. Именно низкая защищенность традиционных языков типа C, C++, Pascal, имеющих «опасные» конструкции типа макросов или указателей, а также отсутствие явной поддержки параллелизма стала одним из побудительных мотивов создания таких языков программирования нового поколения, как D (см. www.digitalmars.com/d/) и Erlang (см. [69] и www.erlang.org). Эти языки еще только начали завоевывать популярность, но заложенные в них новые идеи можно частично реализовать и в используемом для создания пакета CRW-DAQ языке Object Pascal.

Можно сформулировать ряд необходимых условий и принципов безопасного многопоточного программирования, которых придерживался автор. Прежде всего, безопасные программы должны строиться на базе безопасных объектов.

Безопасные программные объекты должны быть верифицируемы, т. е. содержать возможность проверки своей корректности. Например, массивы верифицируемы (индекс элемента можно проверить на диапазон), а указатели – нет (отсутствует механизм их проверки). Они должны быть строго инкапсулированы, т.е. все структуры данных должны быть «спрятаны» за безопасными вызовами процедур и функций. Они должны быть самосогласованны, т.е. не содержать внутренних противоречий. Например, счетчик элементов списка должен совпадать с реальным числом элементов в нем. Они должны быть атомарными, т.е. все доступные процедуры должны неделимым образом переводить объект из одного самосогласованного состояния в другое, тоже самосогласованное, без доступных «промежуточных» состояний. Они должны быть детерминированными, т.е. предсказуемыми, без «побочных эффектов». В частности, они должны быть герметичными, т.е. выделяемые им ресурсы не должны «утекать». По возможности, они должны использовать квазистатическое (однократное) выделение ресурсов, т.к. при динамическом выделении ресурсов есть риск сбоя из-за их нехватки в данный момент. Наконец, желательно, чтобы объекты имели свойство рефлексии, т.е. самонаблюдения. Безопасная программа должна сама тестировать свой код!

Меры повышения надежности можно разделить на три категории – превентивные, восстановительные и диагностические. Превентивные меры ставят

целью предотвратить возникновение ошибок. Главными среди этих мер являются:

- 1) Проверка корректности входных данных перед их использованием. Например, проверка указателя на ноль, проверка индекса массива на допустимый диапазон.
- 2) Использование программных технологий, снижающих риск ошибки. Например, объектно-ориентированное программирование (ООП), «умные указатели», реестр объектов, виртуальная машина (Рис.14), выделение служб в отдельные процессы.

Заметим, что все превентивные меры предполагают, что объекты верифицируемы. Восстановительные меры пытаются ликвидировать или хотя бы снизить последствия уже возникших ошибок. Главной из этих мер является механизм структурной обработки исключений - SEH (Structured Exception Handling), реализуемый в языке Object Pascal при помощи конструкций «try ... except ...» для «перехвата» исключений и «try ... finally ...» для освобождения ресурсов [60].

Диагностические меры, хотя они напрямую не влияют на обработку ошибок, способствуют их своевременному обнаружению, а потому также необходимы для обеспечения программной безопасности. Среди этих мер можно назвать технику счетчиков ресурсов, списков объектов, журнальных файлов, отладочных консолей, гистограмм для сбора статистики и т.д. В совокупности эти меры и обеспечивают рефлексию (самоконтроль) программы.

Автору не удалось найти ни одной открытой программной библиотеки, в которой последовательно выполнялись бы все описанные здесь принципы. Поэтому при создании пакета CRW-DAQ была разработана собственная библиотека функций и классов на языке Object Pascal, и главной её особенностью стала по мере возможности более полная реализация принципов безопасного многопоточного программирования задач мягкого реального времени. Рассмотрим реализацию этой библиотеки более подробно.

Классическая концепция объектно-ориентированного программирования предлагает три основных принципа – инкапсуляцию (объединение данных и кода в одном объекте), полиморфизм (виртуальные функции) и наследование (сохранение методов и данных предка у дочернего класса). С точки зрения безопасности лишь наследование в большинстве языков не вызывает нареканий: повторное использование отлаженного кода в классах снижает риск новых ошибок. А вот реализация полиморфизма в распространенных языках (C++, Object Pascal)

небезопасна, т.к. в ней нет механизма верификации указателей на объекты, и они не проверяются при вызове. В результате, например, вызов виртуального метода для пустого объекта (с указателем *Nil*) приводит к аварийному завершению программы и является распространенной причиной программных сбоев. Инкапсуляция в строгом смысле предполагает, что все данные класса должны быть приватными (*private* или *protected*), а общедоступным (*public*) должен быть только интерфейс класса, то есть только его методы: процедуры, функции и свойства (*property*). Распространенные библиотеки классов (VCL, MFC, ...) не являются строго инкапсулированными, т.к. часто содержат открытые поля данных, не «спрятанные» за методами. Это резко снижает безопасность классов, т.к. без строгой инкапсуляции, имея возможность произвольно менять открытые поля данных, нельзя гарантировать самосогласованности объектов.

Даже соблюдение трех принципов ООП еще не является достаточным условием для безопасного многопоточного программирования. Для реализации безопасных классов требуется более сильное условие – «атомарность», то есть неделимость всех открытых методов объекта. Атомарность, включая строгую инкапсуляцию как необходимое условие, требует также отсутствия у класса «промежуточных», «незавершенных» состояний, чтобы все его доступные извне состояния были самосогласованны. Это значит, что вызов любого метода объекта либо завершается неудачей, возвращая код ошибки, либо неделимым образом переводит объект в новое самосогласованное состояние. Для этого все открытые методы объекта должны быть защищены критическими секциями, мютексами или другими объектами синхронизации потоков. Другим свойством, необходимым для реализации безопасных классов, является «герметичность», т. е. гарантированное отсутствие «утечки» ресурсов при вызове любой функции класса, даже при возникновении в ней исключений. Реализация атомарности и герметичности требует методичного использования техники структурной обработки исключений. Наконец, безопасные классы должны обладать свойством «рефлексии», то есть иметь средства для самодиагностики, проверки корректности ссылок и самосогласованности объектов, вести журнал статистики использования ресурсов.

При реализации атомарных, потокобезопасных объектов используются объекты синхронизации потоков, что создает дополнительные проблемы,

главными из которых являются взаимная блокировка и инверсия приоритета. Взаимная блокировка возникает, когда два потока (А,В) заблокированы на общих ресурсах (С,Д), причем поток А захватил ресурс С и ждет освобождения Д, а поток В захватил ресурс Д и ждет освобождения С, при этом оба потока повисают. Инверсия приоритета возникает в случае трех потоков (А,В,С) с высоким, средним и низким приоритетом. Если поток С захватил ресурс R, а затем поток А заблокировался при попытке его захвата, то поток В может вытеснить более высокоприоритетный поток А на неопределенное время, так как поток В может законно вытеснить низкоприоритетный поток С, который задерживает разблокировку потока А, пока не освободит ресурс R. Наконец, блокировка общих ресурсов создает задержки, приостанавливая выполнение потоков, что также нежелательно. Проблема инверсии приоритетов решается введением механизма наследования приоритетов, когда ОС временно повышает приоритет потока С до уровня потока А, если поток А блокируется на ресурсе R, захваченном потоком С. Однако наследование приоритетов реализовано лишь в ОСРВ, таких как QNX [68], и не реализовано в обычных ОС, включая Windows и Linux. Однако это не значит, что проблема никак не решается в этих системах.

Чтобы минимизировать проблемы, связанные с блокировками ресурсов, в библиотеке CRW-DAQ используется принцип «копируй и пользуйся». Это значит, что для использования какого-то объекта поток блокирует его на минимально возможное время лишь для того, чтобы сделать локальную копию интересующих данных, а затем блокировка снимается и работа происходит с этой копией данных. Например, при прорисовке кривой, которая может занимать значительное время, система визуализации блокирует кривую, копирует видимый фрагмент кривой во временный массив и сразу освобождает кривую. Лишь после этого начинается длительная операция прорисовки кривой. Такой подход позволяет снизить взаимное влияние потоков до минимума, хотя и не снимает проблему полностью.

Одной из главных тем безопасного программирования является контроль использования ресурсов: процессора, памяти, дисков, сетевых каналов и т.д. Вопрос использования процессора решается за счет возможностей приоритетной вытесняющей многозадачности, как было подробно описано выше (Рис.15,Рис.64). Напомним, что задача управления АСКУ в CRW-DAQ всегда разбивается на ряд

параллельно работающих потоков (и процессов), приоритет и период опроса которых регулируется. Принятый в пакете асинхронный стиль программирования также способствует распараллеливанию задачи.

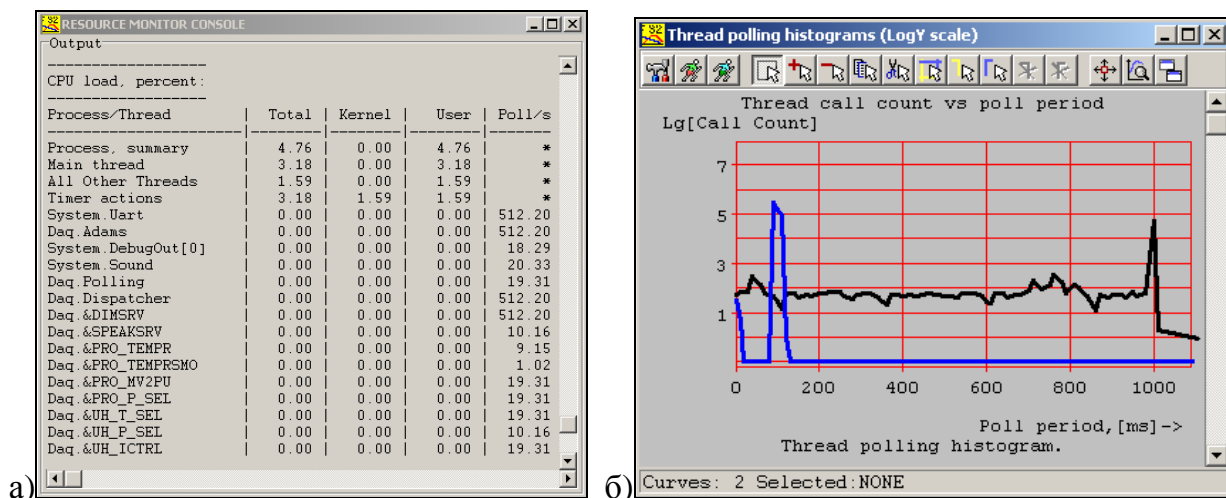


Рис.72. Консоль монитора ресурсов: а) загрузка процессора и частота опроса потоков, здесь - от 1 до 512 раз в секунду; б) гистограммы частоты опроса потоков. Потоки с высоким приоритетом имеют узкий пик, с низким - сильно «размазаны».

Корректная работа с большим числом (обычно более 20) потоков практически невозможна без развитых средств диагностики. Поэтому в пакете имеется набор средств для слежения за ресурсами. Есть монитор ресурсов (Рис.72,а), позволяющий контролировать использование процессора, памяти, дисков и наблюдать частоту опроса всех потоков программы и загрузку процессора по каждому из них. Кроме того, по каждому из потоков постоянно набирается гистограмма периода опроса (Рис.72,б), которая позволяет оценивать временные характеристики системы.

Служба системного времени, используя наличие в компьютере нескольких независимых таймеров, следит за корректностью системных часов и выдает предупреждение, если системное время было изменено. Служба программного сторожевого таймера отслеживает частоту опроса потоков и выдает предупреждение, если какие-то потоки долго не отвечают, то есть, возможно, «повисли». Звуковая система оповещения выдает речевые сообщения в случае ошибок связи с измерительными модулями, так что при обнаружении отклонений поведения программы от нормы оператор получает звуковое или речевое уведомление, что позволяет своевременно принять восстановительные меры.

Наиболее частыми проблемами, связанными с остальными ресурсами (память, диск, сеть), являются нехватка и утечка ресурсов, а также задержка их

захвата. Нехватка ресурсов возникает при неудачной попытке динамического выделения ресурсов после их исчерпания, что приводит к сбою в работе алгоритмов управления. Она может быть спровоцирована не только ошибками в самой АСКУ, но и посторонними ресурсоемкими программами, захватившими требуемые ресурсы. Утечка ресурсов возникает, когда алгоритм для выполнения каких-то действий временно захватывает динамические ресурсы, но потом забывает их освободить. При этом сам ресурс (объект, область памяти, открытый файл) остается занятым, но программа теряет ссылку на него и не может его в дальнейшем использовать. Утечка ресурсов – весьма трудно обнаруживаемая ошибка, поскольку она имеет латентный период (проявляется не сразу): АСКУ может сутками работать без видимых ошибок. Однако утечка ресурсов обладает кумулятивным свойством: со временем утечки накапливаются и неизбежно ведут к исчерпанию ресурсов, после чего следует неизбежный сбой управления из-за их нехватки. Задержка захвата ресурсов играет роль в алгоритмах управления реальным временем, ведь даже если свободные ресурсы есть, их поиск и захват может занять слишком много времени.

Перечислим меры борьбы с ошибками ресурсов в пакете CRW-DAQ. Однократный (квазистатический) захват ресурсов является хорошей превентивной мерой против их нехватки, утечки и задержки, поэтому большинство необходимых для работы АСКУ ресурсов выделяется при загрузке её конфигурации и освобождается при её завершении. Хотя это явно неоптимальное использование ресурсов, так как не все захваченные ресурсы могут потребоваться в конкретном сеансе работы АСКУ, но квазистатический захват ресурсов действительно снимает проблему их нехватки (они уже захвачены), утечки и задержки (в процессе работы АСКУ они не захватываются и не освобождаются). Конечно, квазистатический захват ресурсов требует более тщательного конфигурирования АСКУ, чтобы однократно выделенных ресурсов хватало для решения данной задачи управления. Однако после такого конфигурирования АСКУ становится более предсказуемой: она либо не загрузится вовсе, если ресурсов мало, либо загрузится и будет гарантированно работать сколь угодно долго без проблем с ресурсами. Другой мерой борьбы с утечками ресурсов являются счетчики: в каждую функцию захвата и освобождения ресурсов, а также в конструкторы и деструкторы классов

включаются процедуры инкремента и декремента глобальных счетчиков. Много времени это не требует, однако дает возможность в любой момент оценить интегральное количество выделенных ресурсов. Ненулевое значение счетчиков при завершении программы, когда все объекты должны быть уже уничтожены, является явным признаком утечки ресурсов, поэтому все счетчики перед завершением программы записываются в журнальный файл для последующего анализа. Можно сказать, что фактически все время работы программы идет ее самотестирование. Хотя счетчики - чисто диагностическое средство, они позволяют своевременно найти и затем устранить утечки ресурсов. Наконец, в качестве восстановительных мер борьбы с утечками ресурсов применяются методы структурной обработки исключений.

Другой частой причиной ошибок являются «битые ссылки», то есть некорректные указатели на объекты. Например, если объект не был создан или его ссылке забыли присвоить значение, то возникает пустая ссылка (ноль или *Nil*). Если при уничтожении объекта его ссылку забыли очистить, возникает «мертвая ссылка» на уже несуществующий объект. При попытке использования битой ссылки объекта для вызова его методов зачастую возникает аварийный сбой программы. Дело усугубляется тем, что в современных компиляторах указатели не верифицируемы, поэтому способа проверки корректности ссылок нет. Оператор «*is*» не спасает: он может установить класс объекта только для корректных ссылок. В многопоточной программе объект может к тому же одновременно использоваться и уничтожаться разными потоками, что часто и служит причиной появления мертвых ссылок.

Для предотвращения ошибок, связанных с битыми ссылками, приняты следующие меры. Во-первых, во всех методах классов используется тест корректности ссылки объекта выражением «*if Assigned(Self) then ...*» на его входе. Эта проверка, занимая один такт процессора, практически не снижает скорости, но предохраняет от ошибки в случае вызова функции для пустой ссылки (*Nil*) объекта. Так можно защитить только вызовы статических методов, но не виртуальных, использующих указатель *Self* для получения адреса вызываемого метода. Поэтому для безопасной реализации класса все его виртуальные методы должны быть скрытыми (*protected*), а в открытом интерфейсе должны присутствовать только статические методы и

свойства (*property*), которые можно использовать в качестве «обертки» для скрытых виртуальных методов (11, Листинг 5).

Таким образом, сохраняя полиморфизм (виртуальные методы), открытый интерфейс объекта остается безопасным даже при использовании пустых ссылок.

Проблема мертвых ссылок решалась автором с помощью «умных ссылок», идея которых состоит в следующем. В объекте заводится поле *Master*, хранящее адрес умной ссылки - статической переменной, содержащей указатель на объект. При создании объекта его адрес присваивается умной ссылке, а сам адрес умной ссылки запоминается в поле *Master*. Деструктор объекта перед его уничтожением проверяет *Master*, и если он не нулевой, очищает умную ссылку: «*Master^:=Nil*». В результате, когда бы ни был уничтожен объект, он сам очистит ссылку на себя. Поэтому, при использовании описанного метода, мертвых ссылок никогда не возникает, так как умная ссылка всегда или корректна, или пуста (Листинг 6).

Легко заметить, что сочетание умных ссылок и защиты от пустых указателей резко снижает риск ошибок: умные ссылки гарантируют, что ссылка корректна или пуста, а защита гарантирует безопасность пустых ссылок.

Рассмотрим теперь способ реализации безопасной функций класса (Листинг 7), которого старался придерживаться автор при создании классов пакета CRW-DAQ. В начале функции ей присваивается некоторое значение «по умолчанию», например, ноль. Потом проверяется корректность ссылки объекта «*if Assigned(Self) then ...*» для защиты от пустых (*Nil*) ссылок (для них возвращается значение по умолчанию). Далее, тело функции помещается в защищенную секцию «*try ... except ... end*», чтобы перехватывать возникающие в нем исключения. Внутри защищенной секции вложена конструкция «*Lock; Allocate; try ... finally Deallocate; Unlock; end*». Пара процедур «*Lock-Unlock*» обеспечивает потоковую безопасность (блокировку и разблокировку), например, с помощью критических секций, а пара процедур «*Allocate-Deallocate*» обеспечивает захват и освобождение ресурсов. Эта структура функции гарантирует освобождение всех занятых ресурсов и возврат разумного значения даже при возникновении исключения в теле процедуры «*Work*», выполняющей интересующие вычисления. Построенные таким образом функции согласуются с принципами ООП, SEH, атомарности и герметичности.

Следующим механизмом, обеспечивающим безопасность классов, являются счетчики и списки объектов классов, идея которых состоит в следующем. Для каждого класса заводится глобальный счетчик и/или список объектов этого класса. Каждый конструктор класса содержит процедуру инкремента счетчика и/или регистрации созданного объекта в списке, а деструктор обеспечивает декремент и/или удаление объекта из списка перед его уничтожением. В результате в каждый момент времени известно число и/или полный список объектов класса, что дает возможность контролировать корректность ссылок и утечки памяти. В частности, в конце работы программы, когда все объекты должны быть уничтожены, наличие ненулевых счетчиков является признаком утечки ресурсов и регистрируется в журнальном файле для последующего анализа (Листинг 8).

Описанная выше техника счетчиков и списков соответствует принципу рефлексии: пакет фактически постоянно тестирует корректность своего кода, ведя журнал ошибок и использования ресурсов. Анализ этого журнала позволяет диагностировать трудно выявляемые редкие ошибки и тем самым способствует повышению надежности пакета. Списки, кроме того, делают указатели объектов верифицируемыми. Отметим также, что счетчики обычно изменяются парой «атомарных» функций «*InterlockedIncrement*» и «*InterlockedDecrement*», что требуется для корректной работы в многопоточной среде.

Заслуживает внимания реализованный в пакете глобальный реестр объектов. Все классы в пакете CRW-DAQ наследуются от *TMasterObject*, в который встроен механизм регистрации объектов в глобальном списке – реестре объектов. В каждом объекте есть поле *TMasterObject.Ref* для индекса объекта в реестре, содержащего массив указателей на объекты. Конструктор и деструктор автоматически регистрируют объекта в реестре и удаляют его, что позволяет поддерживать взаимнооднозначное соответствие указателей и индексных ссылок. В реестре есть 3 основные операции: индексация (получение указателя по индексу), вставка и удаление. Индексация является быстрой операцией порядка $O(1)$. Вставка и удаление для обычного списка (непрерывный массив со счетчиком) размера N имеет порядок $O(N)$, что слишком медленно, но можно повысить скорость за счет комбинации массива указателей и стека для хранения его свободных индексов. В начальный момент все индексы заносятся в стек, имеющий тот же размер, что

массив указателей. При вставке объекта в реестр из стека берется свободный индекс, по индексу в массив пишется указатель объекта, а сам индекс заносится в поле *Ref* объекта. При удалении объекта ссылка по индексу *Ref* очищается, а индекс возвращается в стек. В результате все операции имеют порядок $O(1)$, т.е. не зависят от размера реестра. Для иллюстрации (Листинг 9) приведена сильно урезанная реализация реестра объектов, показывающая принцип его работы (для простоты опущена потоковая защита и динамический размер реестра).

Индексные ссылки объектов в реестре верифицируемы и безопасны, а по индексу можно быстро получить указатель объекта. Поэтому главной целью введения реестра объектов было создание безопасных библиотек, работающих с индексными ссылками вместо указателей. Индексы объектов используются, например, в качестве ссылок в прикладных программах DAQ Pascal для работы с различными объектами АСКУ: тегами, кривыми, устройствами, процессами, окнами и т.д. Использование верифицируемых индексных ссылок вместо указателей позволяет резко снизить вероятность серьезных ошибок в прикладных алгоритмах и повысить тем самым надежность программного обеспечения АСКУ.

Принцип виртуальной машины (Рис.14), реализованный автором во встроенном языке прикладного программирования DAQ Pascal, также повышает надежность прикладных программ, как было подробно описано выше.

Автором создана большая библиотека функций измерения времени, ввиду их важности для задач управления. В современных компьютерах есть несколько независимых источников времени: кварцевый генератор на материнской плате, тактовый генератор процессора, энергонезависимые часы CMOS, сетевое время Internet. Есть несколько систем счета времени: астрономическое, календарное, от начала сеанса и т.д. Измерение времени в CRW-DAQ основано на следующих системных вызовах Windows. Функция *GetTickCount* дает 32-битный счетчик миллисекунд от старта системы, инкрементируемый примерно каждые 10 мс по прерываниям таймера. Счетчик меняется монотонно и непрерывно, но через 49.7 дней переполняется, поэтому напрямую непригоден для измерения времени. Функция *GetTimeAsFileTime* дает линейное время в 100-нс отсчетах от 01.01.1601, имеет тот же квант времени ~10 мс. Её недостаток в том, что время может меняться скачкообразно и не быть монотонным, что происходит, например, в момент

изменения системного времени. Это может нарушить работу алгоритмов реального времени, поэтому в явном виде её тоже использовать нельзя. Основной функцией измерения астрономического времени в пакете CRW-DAQ стала *msecnow*, время в миллисекундах от начала эры, использующая комбинацию *GetTimeAsFileTime* для фиксации начала отсчета и *GetTickCount* для приращения времени от начала отсчета, с учетом переполнения счетчика. Её свойства – линейность, непрерывность и монотонность времени, быстрый вызов (~0.1 мкс), квантование времени с шагом ~10 мс. Комбинация вызовов *QueryPerformanceCounter*, *QueryPerformanceFrequency* позволяет измерять время с высоким разрешением, зависящим от конкретной аппаратуры. На них основана функция *mksecnow*, число микросекунд от старта программы. Её достоинство состоит в линейности, монотонности и высоком разрешении (~1 мкс), недостаток в более высокой стоимости вызова (~1 мкс) и аппаратной зависимости. Очень быстрым (~30 нс на вызов) способом измерения времени является команда RDTSC чтения счетчика тактов процессора (Time Stamp Counter). Она позволяет измерять очень короткие интервалы времени (от 30 нс), а также оценивать производительность программного кода (в тактах процессора). Её недостатком является зависимость от частоты процессора, которую еще надо определить, и невозможность использования функции при переменной частоте тактов, применяемой в современных процессорах для снижения энергопотребления, что делает её аппаратно зависимой. Есть также ряд специфических функций, таких как *RunCount* (счетчик вызовов программы) или *vdpm_count* (счетчик операций DAQ Pascal). Большое число функций времени позволяет эффективно решать задачи управления АСКУ в реальном времени: измерять и задавать интервалы, генерировать периодические события по расписанию, контролировать объем вычислений.

Говоря об особенностях пакета, надо упомянуть о наличии средств симуляции объектов управления. При разработке прикладных программ в пакете активно используются симуляторы, то есть программы, моделирующие работу системы сбора данных для отладки алгоритмов управления без подключения реальной аппаратуры. Симуляторы являются очень важным и подчас незаменимым инструментом для разработки АСКУ, ведь далеко не всегда существует возможность проверки всех алгоритмов управления на реальной аппаратуре, даже

если она подключена и работает. Например, бывает затруднительно реализовать условия для проверки системы блокировки в случае радиационной аварии, и здесь симуляторы оказываются незаменимыми.

Симуляторы обычно реализуются в виде набора конфигурационных файлов и программ на языках DAQ Script, DAQ Pascal, которые временно (при отладке) подключаются вместо измерительных модулей, моделируя их работу. Возможность редактирования и компиляции программ в процессе измерений позволяет легко менять алгоритм симуляции «на ходу», что ускоряет отладку. Наличие средств симуляции – важное средство повышения надежности прикладных программ, поскольку они позволяют проверять работу алгоритмов управления в самых жестких условиях, трудно воспроизводимых на реальной аппаратуре.

11. Листинги

Листинг 1. Структура программного интерфейса CrwApi.

```
unit _CrwApi;
interface
type
TCrwApi = class; // Корень дерева интерфейсов CRW API.
TSysApi = class; // API для функций общего назначения.
TGuiApi = class; // API для создания графического интерфейса.
TDaqApi = class; // API для систем сбора данных.
TDanApi = class; // API для анализа данных.
TCrwApi = class(TObject)
function Version: Cardinal; virtual; abstract; // Версия API
function SysApi: TSysApi; virtual; abstract;
function GuiApi: TGuiApi; virtual; abstract;
function DaqApi: TDaqApi; virtual; abstract;
function DanApi: TDanApi; virtual; abstract;
end;
TGuiApi = class(TObject)
// Вывод текста в консольное окно.
procedure Echo(const Msg: String); virtual; abstract;
... Другие методы ...
end;
... Другие классы ...
implementation
end.
```

Листинг 2. Пример вызова DLL библиотеки из программы на языке DAQ Pascal.

```
program DllDemo;
var handle, cmd: Integer;
begin
cmd:=1;
handle:=DaqDllInit('c:\demo\test.dll');
if handle=0 then writeln('Error!') else
if DaqDllCall(handle, DAQ_CMD_POLL)
then writeln('Success DLL call')
else writeln('Could not call DLL!');
DaqDllFree(handle);
end;
```

Листинг 3. Пример запуска дочернего процесса и обмена данными с ним.

```
program TaskDemo;
var line:string; tid:Integer;
begin
  if RunCount=1 then begin
    tid:=task_init('cmd.exe');
    if task_run(tid)
    then writeln('cmd.exe is running now.')
    else writeln('Could not run cmd.exe!');
  end else
  if IsInf(RunCount) then begin
    if task_free(tid) then tid:=0;
  end else begin
    if not eof then begin
      readln(line);
      if task_send(tid,line)
      then writeln('Data sent:',line);
    end;
    line:=task_recv(tid,255);
    if Length(Line)>0 then writeln('Data received:',line);
  end;
end.
```

Листинг 4. Пример программы обмена данными по сетевому каналу.

```
// Инициализация может быть разной:
p:=pipe_init('task cmd.exe /c dir'); // Открыть анонимный канал связи
p:=pipe_init('pipe daqhost\datapipe'); // Открыть именованный канал связи
p:=pipe_init('tcp port 123 server'); // Открыть TCP/IP порт 123 как сервер
p:=pipe_init('tcp port 123 client'); // Открыть TCP/IP порт 123 как клиент
p:=pipe_init('com port 1 baudrate 9600'); // Открыть RS-232 порт COM1
// Но дальнейшая работа с каналом будет одинаковой:
if pipe_run(p) then writeln('Error!'); // Запуск дочернего процесса task
// Серверная часть:
if pipe_connected(p)>0 then // Если канал открыт
for i:=0 to pipe_count(p)-1 do begin // Цикл по клиентам
  ps:=pipe_stream(p,i); // Взять i-й клиентский поток
  if pipe_connected(ps)>0 then begin // Если он открыт
    Data:=pipe_recv(ps,255); // Читать данные
    pipe_send(ps,Data); // Записывать данные
  end;
end;
// Клиентская часть:
if pipe_connected(p)>0 then begin // Если канал открыт
  Data:=pipe_recv(p,255); // Читать данные
  pipe_send(p,Data); // Записывать данные
end;
pipe_free(p); // Закрыть канал связи
```

Листинг 5. Пример объявления защищенного от «пустых» ссылок класса.

```
TDemo = class(TObject) // Пример защищенного класса
private // Закрытые (приватные) поля:
  myData:Integer; // Скрытое поле данных
private // Закрытые (приватные) методы:
  function GetData:Integer; // «Обертка» для поля данных
protected // Закрытые виртуальные методы
  function DoGetData:Integer;virtual; // Виртуальная функция чтения данных
public // Открытые (публичные) поля:
  constructor Create(d:Integer);
  property Data:Integer read GetData; // Открытый интерфейс для чтения данных
end;
function TDemo.GetData:Integer; // Защищенная функция - «обертка»
begin
  Result:=0; // Значение по умолчанию
  if Assigned(Self) // Если ссылка корректна
  then Result:=DoGetData; // Тогда вызвать виртуальный метод
end;
```

```

function TDemo.DoGetData:Integer; // Виртуальная функция чтения данных
begin
    Result:=myData; // Например, просто чтение поля myData
end;
constructor TDemo.Create(d:Integer); // Конструктор объекта
begin
    inherited Create;
    myData:=d;
end;
// Пример использования:
program Test;
var p:TDemo=nil; // Ссылка объекта с инициализацией при старте
begin
    i:=p.Data; // Результат i=0; вызов корректен, хотя p=nil
    p:=TDemo.Create(10); // Создание объекта
    i:=p.Data; // Результат i=10
end.

```

Листинг 6. Пример защищенного класса с использованием «умных» ссылок.

```

TDemo = class(TObject) // Пример защищенного класса
private // Закрытые (приватные) поля и методы:
    myMaster:^TObject; // Скрытое поле адреса «умной ссылки»
    procedure SetMaster(var M:TObject); // Присвоение «умной ссылки»
public // Открытые (публичные) методы:
    constructor Create;
    destructor Destroy; override;
    property Master:TObject write SetMaster; // Установка «умной ссылки»
end;
constructor TDemo.Create; // Конструктор объекта
begin
    inherited Create; // Создание объекта
    myMaster:=nil; // Очистка указателя «умной ссылки»
end;
destructor TDemo.Destroy; // Деструктор объекта
begin
    if myMaster<>nil // Если адрес «умной ссылки» назначен,
    then myMaster^:=nil; // То очистить «умную ссылку» объекта
    inherited Destroy; // Уничтожить объект
end;
procedure TDemo.SetMaster(var M:TObject); // Установка «умной ссылки»
begin
    if Assigned(Self) then myMaster:=@M; // Запомнить адрес «умной ссылки»
end;
// Пример использования:
program Test;
var p:TDemo=nil; // Ссылка объекта с инициализацией при старте
begin
    p:=TDemo.Create; // Создание объекта
    p.Master:=p; // Присвоение «умной ссылки»
    ... // Использование объекта
    p.Free; // Уничтожение объекта
    // При уничтожении «умная ссылка» p обращается в ноль: p=Nil!
end.

```

Листинг 7. Пример атомарного и защищенного метода класса.

```

function TDemo.Test:Integer; // Пример защищенной функции класса
begin
    Result:=0; // Задаем значение по умолчанию
    if Assigned(Self) then // Тест корректности ссылки объекта
    try // Защищенная секция try ... except ...
        Lock; // Блокировка объекта
        Allocate; // Выделение нужных ресурсов
        try // Защищенная секция try ... finally ...
            ... Work; ... // Выполнение необходимых действий
            Result:=...; // Окончательный результат вызова
        finally // Секция освобождения ресурсов
            Deallocate; // Освобождение ресурсов
            Unlock; // Разблокировка объекта
        end;
    end;
end;

```

```

    end;
except
    on E:Exception do Report(E.Message); // Секция обработки исключения:
    // Например, сообщение об ошибке
end;
end;
end;

```

Листинг 8. Пример классов с использованием счетчиков и списков объектов.

```

unit Demo; // Модуль с описанием класса
interface uses Classes;
const
    DemoCount:Integer=0; // Глобальный счетчик объектов класса
    DemoList:TList=Nil; // Глобальный список объектов класса
type
    TDemo = class(TObject) // Объявление класса
    public // Открытые (публичные) поля:
        constructor Create); // Создать объект
        destructor Destroy; virtual; // Уничтожить объект
    end;
implementation
    constructor TDemo.Create; // Создать объект
    begin
        inherited Create;
        InterlockedIncrement(DemoCount); // Поточно-безопасный инкремент счетчика
        DemoList.Add(Self); // Регистрация объекта в списке
    end;
    destructor TDemo.Destroy; // Уничтожить объект
    begin
        DemoList.Delete(Self); // Удаление объекта из списка
        InterlockedDecrement(FifoCount); // Поточно-безопасный декремент счетчика
        inherited Destroy;
    end;
initialization
    DemoCount:=0; // Обнуляем счетчик при старте программы
    DemoList:=TList.Create; // Создаем список при старте программы
finalization
    if DemoCount<>0 // Если счетчик в конце ненулевой
    then writeln('Утечка:',DemoCount); // То пишем в журнал факт утечки ресурсов
end.
Program Demo; // Демонстрация использования списков
Uses Demo; var p:TDemo;
begin
    p:=TDemo.Create; // Создаем объект
    ...
    if DemoList.IndexOf(p)>=0 // Проверяем, существует ли такой объект
    then ... // и только тогда используем.
    writeln(DemoCount); // Выводим счетчик числа объектов TDemo
    ...
end;

```

Листинг 9. Упрощенная реализация реестра объектов – иллюстрация принципа его работы.

```

program ObjectRegistryDemo; // Иллюстрация принципа реестра объектов
var reg : array[1..1024] of TObject; // Реестр объектов - массив указателей
    stack : array[1..1024] of Integer; // Стек свободных индексных ссылок
    sp : Integer; // Указатель стека индексных ссылок
    obj : TObject; // Демонстрационный объект
    ref,i : Integer; // Демонстрационная ссылка
procedure push(ref:Integer);
begin sp:=sp+1; stack[sp]:=ref; end; // Процедура кладет индекс в стек
function pop:Integer;
begin pop:=stack[sp]; sp:=sp-1; end; // Функция берет индекс из стека
type
    TDemo=class(TObject) // Объект, регистрирующий себя в реестре
    ref:Integer; // Индексная ссылка в реестре
    constructor Create;
    destructor Destroy; override;
end;
constructor TDemo.Create;
begin

```



```

    inherited Create;
    ref:=pop; // Берем из стека свободную ссылку
    reg[ref]:=Self; // Кладем указатель объекта в реестр
end;
destructor Destroy;
begin
    reg[ref]:=Nil; // Очищаем указатель в реестре
    push(ref); // Возвращаем ссылку в стек
    inherited Destroy;
end;
begin
    sp:=0; while sp<1024 do push(sp+1); // Инициализация стека
    obj:=TDemo.Create; // Создание объекта
    ref:=obj.ref; // Запоминание ссылки объекта
    ...
    if (ref>0) and (ref<=1024) then // Верификация индексной ссылки
    if reg[ref] is TDemo then // Верификация указателя объекта
    with reg[ref] as TDemo do ... // Использование объекта по ссылке
    ...
    reg[ref].Free; // Уничтожение объекта по ссылке
    ref:=0; // Очистка ссылки
    ...
    if sp<>1024 // Если в конце не все индексы в стеке
    then writeln('Memory leak!'); // то есть утечка памяти-неубитые объекты
end.

```