

# Использование ОС QNX 4 в проектах автоматизации стендовых испытаний ОАО «УМПО»

Леонид Хаит, Аркадий Выдрин, Михаил Головин, Руслан Ямалов, Вадим Солдатов

В статье рассказывается об автоматизированной информационно-измерительной системе, функционирующей на стендах ОАО «УМПО» (г. Уфа). Система предназначена для настройки и испытания авиационных двигателей семейства «СУ».

Мы не успели оглянуться, как пролетело 12 лет с момента публикации нашей первой статьи в журнале «СТА» – «Автоматизированные информационно-измерительные системы испытаний авиационных двигателей» (№ 1 за 1999 год). Там мы описали первый опыт использования ОС QNX 4 в автоматизированной системе испытаний (АИИС) авиационного двигателя.

В настоящее время на испытательных стендах нашего предприятия для испытания двигателей гражданского и военного назначения и их узлов уже много лет работают АИИС с ОС QNX 4. Теперь можно сделать некоторые выводы и попытаться построить планы на будущее.

## СТРУКТУРНАЯ СХЕМА СИСТЕМЫ

Система имеет одноуровневую архитектуру (рис. 1). Промышленные компьютеры (контроллеры) объединяют в себе функции сбора, измерения и управления процессами испытания, а также выступают в роли автоматизированных рабочих мест. Это стало возможным за счёт применения ОС реального времени QNX.

## ОС QNX4 НАС НЕ РАЗОЧАРОВАЛА

Операционная система QNX 4 по-прежнему остаётся удобной средой для создания АИИС ответственных применений. QNX 4 работает надёжно. На некоторых жёстких дисках эта ОС была установлена более 10 лет назад и не

требует переустановки. Большинство проблем, возникающих при эксплуатации АИИС, связаны с нашими собственными ошибками в прикладных программах либо с возникающими отказами аппаратной части.

Достоинства ОС QNX 4 – микроядерная архитектура, а также поддержка жёсткого реального времени, встроенной сети, механизма обмена синхронными сообщениями между процессами. Они заложены в основу прикладных программных решений АИИС.

## ДВА СЕРВЕРА, МНОГО КЛИЕНТОВ

При старте прикладного программного обеспечения (ПО) в первую очередь запускаются программы-серверы:

- сервер параметров АИИС **db\_param**;
- сервер данных АИИС **receiver**.

Сервер параметров работает с базой данных параметров АИИС, в которой описан каждый измеряемый или вычисляемый параметр системы, указаны его имя в системе, имя и формат для индикации, диапазоны и цвета аварийных и предаварийных значений, описаны способ взаимодействия с аппаратурой в случае его измерения и метрологические характеристики (градуировочные коэффициенты полиномов, характеристики термопар и т.д.).

В качестве СУБД мы используем Raima Database Manager (RDM 4.5), которая позволяет описывать гибкую (табличную и сетевую) структуру данных, а также работает абсолютно надёжно.

Сервер данных **receiver** в момент старта отправляет запросы в **db\_param** и настраивается на тот состав параметров, который определён для текущих испытаний и зависит от типа испытываемого двигателя и вида испытаний.

Обмен данными между **receiver** и **db\_param**, как и между всеми программами АИИС, выполняется посредством встроенного в QNX 4 механизма синхронных сообщений Send-Receive-Reply (SRR). Для организации обмена сообщениями в общих для всех программ **h-файлах** описаны структуры сообщений, в которых первые байты указывают на тип сообщения, а последующие данные представляют содержание сообщения.

Таким образом, **receiver**, отправив в **db\_param** сообщение с кодом запроса «Отправь мне все главные записи параметров АИИС», в синхронном ответе получает число параметров в текущей конфигурации АИИС и описание всех этих параметров.

По такому же принципу непрерывно и с высокой частотой (от нескольких десятков до 1000 сообщений в секунду) обмениваются данными все прикладные программы АИИС.

## ДЕКОМПОЗИЦИЯ ПРОГРАММНОГО ПРОЕКТА

Механизм мгновенных SRR-сообщений позволил разбить большой программный проект на много отдельных программ (процессов), каждая из которых отвечает за отдельную функцию АИИС. Таких программ



Внешний вид кабины наблюдения испытательного стенда

обычно более трёх десятков, в основном они разделяются на программы обслуживания аппаратуры и программы сбора, обработки, визуализации, протоколирования измерений и управления технологическими системами стенда.

Запустив программы-серверы **db\_param &** и **receiver &**, мы создаём среду функционирования любой прикладной программы проекта. Любую другую программу можно отдельно запускать или выполнять её пошаговую отладку.

Таким образом, весьма сложные и объёмные программные проекты АИ-ИС (сотни тысяч строк программного кода) разбиты на отдельные прикладные программы разумного размера, которые легко можно добавлять, дорабатывать и отлаживать, «не задевая» работу других прикладных программ проекта.

В целом такой подход позволяет создавать программные проекты любой сложности путём наращивания количества отдельно запускаемых прикладных программ небольшого размера,

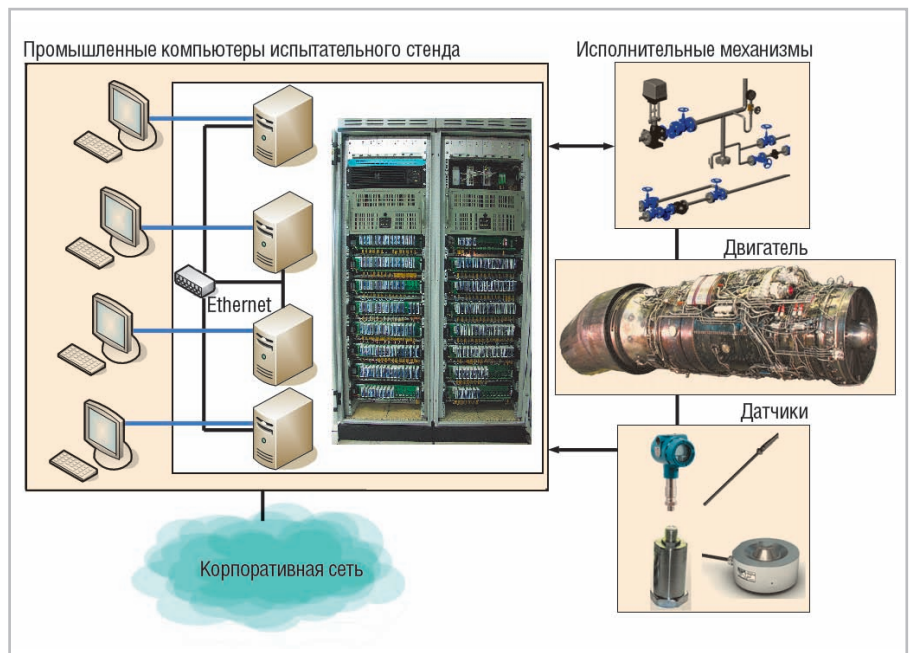


Рис. 1. Структурная схема системы

которые реализуют нужные функции АИИС.

Все программы обмениваются между собой данными посредством механизма синхронных сообщений SRR.

### ПРОГРАММЫ ОБСЛУЖИВАНИЯ ИЗМЕРИТЕЛЬНОЙ АППАРАТУРЫ

Производители различных плат (АЦП, ЦАП, интерфейсов дискретного ввода-вывода, интерфейсов специализированных цифровых каналов пе-



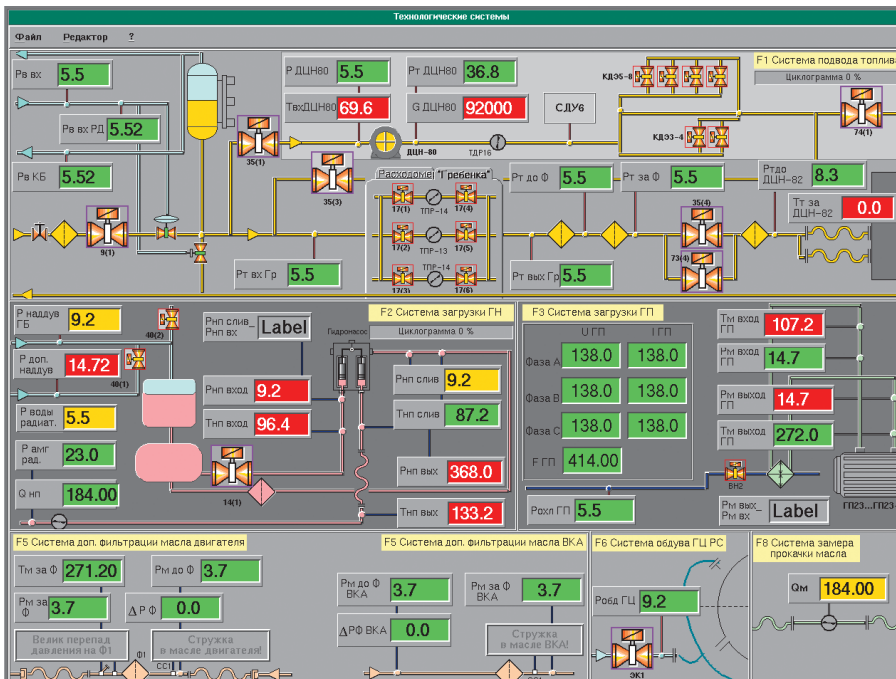


Рис. 2. Мнемосхема технологических систем

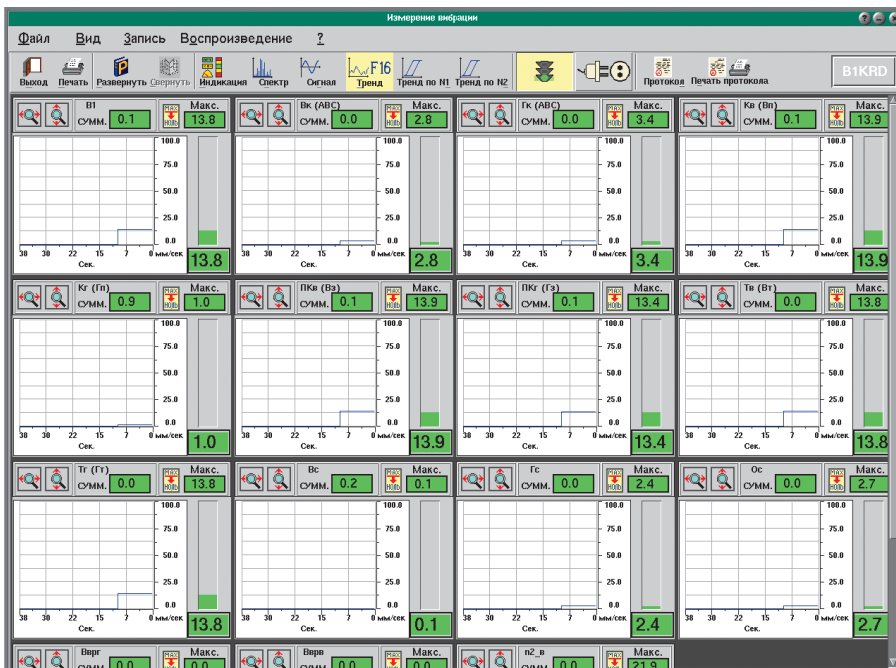


Рис. 3. Экран программы Wvibro

редачи данных) обычно поставляют драйверы для ОС Windows, которые нам не подходят.

Мы пошли по пути написания собственных прикладных программ, которые работают с аппаратурой, а результаты измерений отправляют в сервер данных receiver.

Для того чтобы прикладная программа могла работать с аппаратурой в ОС QNX 4, её нужно собрать с ключом T1 и запускать с правами суперпользователя. Конечно, это более трудоёмкий путь: нужно разобраться с портами ввода-вывода, внутренними буферами

устройства и научиться обрабатывать аппаратные прерывания. Зато нам удаётся работать с измерительными платами с максимальной эффективностью.

Например, прикладная программа **m\_analog** умеет работать с платами АЦП PCI-1713, PCL-816, RTI-820 и со специализированным АЦП L-1221. Если в контроллере АИИС установлено несколько плат АЦП разного типа, то программа **m\_analog** запускается несколько раз, а в ключах запуска указывается базовый адрес, вектор прерывания и тип платы АЦП.

Следует отметить, что некоторые отечественные производители (FA-STWEL, ЭЛКУС) поставляют драйверы для ОС QNX.

Программы обслуживания измерительной аппаратуры работают в фоновом режиме и не заметны для оператора АИИС.

Механизмы диспетчеризации ОС QNX 4 позволяют распределять ресурсы контроллера между процессами так, что каждая программа имеет достаточно возможностей, чтобы обслужить своё устройство. Чаще всего мы запускаем программы с приоритетом по умолчанию 10, и QNX успешно распределяет между ними аппаратные ресурсы контроллера.

И только изредка, когда число аппаратных прерываний превышает 150 в секунду, мы повышаем приоритет прикладной программы до значения 14.

В вопросах поддержки различных аппаратных средств действенную помощь и поддержку в России оказывает фирма ООО «СВД ВС» из Санкт-Петербурга.

### «ПЛОСКАЯ» АРХИТЕКТУРА АППАРАТНЫХ СРЕДСТВ

ОС QNX имеет уникальную возможность обмениваться сообщениями как между программами одного контроллера, так и между программами, работающими на разных контроллерах. По большому счёту, прикладной программе не важно, получает ли она сообщение от соседней программы, или они к ней приходят через сеть.

Объединив несколько контроллеров Ethernet-сетью, мы получили распределённую измерительно-вычислительную среду, все ресурсы которой доступны прикладным программам, функционирующим на любом контроллере в сети. Эти же контроллеры выполняют визуализацию результатов измерения и обработки на видеомониторах АИИС.

Таким образом, благодаря ОС QNX мы не разбиваем аппаратные средства на нижний и верхний уровень, а совмещаем функции измерения и управления с функциями визуализации в нескольких равноправных контроллерах, соединённых сетью.

Число таких контроллеров чаще всего определяется аппаратной «начинкой» средств аналогового и дискретного ввода-вывода и числом рабочих мест операторов АИИС.

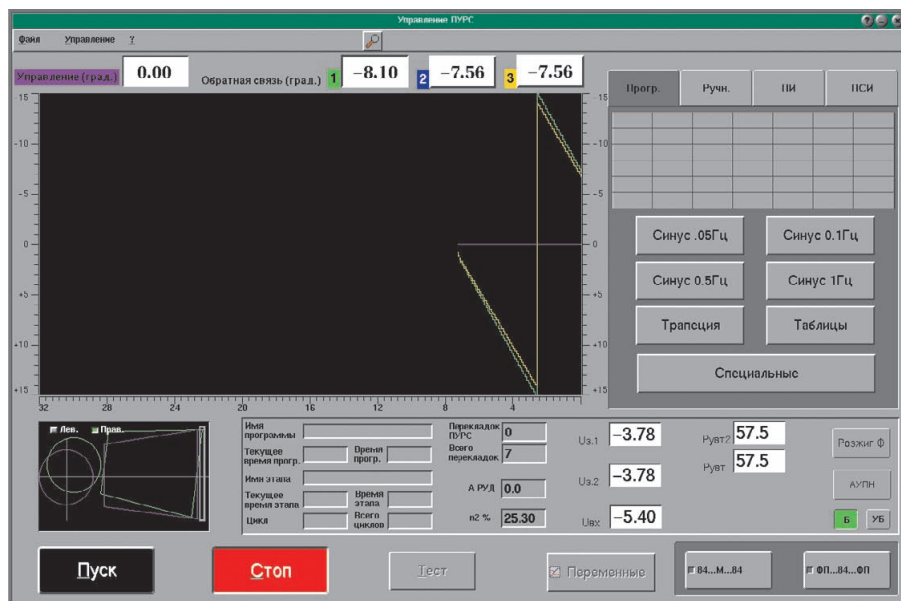


Рис. 4. Экран программы Soplo

### СИНХРОНИЗАЦИЯ ИЗМЕРЕНИЙ НА ПРОГРАММНОМ УРОВНЕ

Нам не требуются аппаратные средства синхронизации измерений, выполняемых на каждом отдельном контроллере. При старте АИИС с помощью системных вызовов QNX 4 синхронизируется текущее время между контроллерами с точностью до 1 миллисекунды. Данные измерений с разных контроллеров с помощью механизма синхронных сообщений SRR попадают в программу-сервер данных **receiver** и в тот же момент приводятся к единому времени контроллера, где работает программа **receiver**.

### ПРОГРАММЫ ВИЗУАЛИЗАЦИИ

Программы визуализации реализуют основные функции АИИС по обработке, представлению данных измерений, их протоколированию и непрерывной записи параметров двигателя и стенда в ходе испытаний. К ним относятся программы обработки параметров статических и динамических режимов работы двигателя, программы обработки сигналов вибрации, программа управления поворотным устройством реактивного сопла по заданным циклограммам и множество других программ.

Для визуализации мы используем графическую среду Photon 1.14, которая входит в состав ОС QNX 4.

### Программа Mmemo

Эта программа используется для работы с технологическими системами. Она отображает на экране различные

мнемосхемы (рис. 2), которые создаются в PhAB – встроенной среде разработки интерфейсных приложений. Созданная мнемосхема настраивается в программе **Mmemo**, в которой элементам индикации назначаются связи с параметрами, а элементам управления – необходимые варианты обработки событий.

Программа позволяет отображать состояние стендовых систем в анимированном виде, делая это более понятным для оператора. Цветовая сигнализация отображаемых параметров привлекает внимание к аварийно-опасным значениям.

Управление электрическими кранами, клапанами и переключателями осуществляется посредством сенсорного экрана либо манипулятора мышь. Также программа позволяет подключать управляющие программы для автоматизации управления и предотвращения аварийных ситуаций.

### Программа Wvibro

В ходе испытания двигателей положено вести постоянное измерение и обработку вибрации двигателя и его систем. Для этого нами была создана программа **Wvibro** (рис. 3). Программа отображает в ходе испытания тренды изменения значений вибрации, позволяет просматривать форму сигналов, по которой можно судить об исправности вибродатчика, строит спектры сигналов, по которым можно сделать вывод о вкладе в вибрацию ротора низкого давления и ротора высокого давления двигателя и вести расчёт разложения вибрации по роторным гармоникам. Для более детального анализа нами применяются математические

программные пакеты DIAdem и Mat- Lab. Для удобства работы с этими программами был выбран формат записи данных TDMS.

### Программа Soplo

Для испытания уникального в мировом двигателестроении устройства поворотного реактивного сопла была создана программа **Soplo** (рис. 4). Программа проводит выполнение заданных циклограмм на предъявительских и приёмо-сдаточных испытаниях, выдавая в режиме реального времени управляющие команды в ЦАП с частотой 1000 Гц.

### Программы Fast и Wpoint

Во время испытания двигателя необходимо оценить его поведение на установившихся и переходных режимах работы. Оценка работы двигателя на переходных режимах делается программой **Fast**, а на установившихся режимах – программой **Wpoint**. Эти программы в режиме реального времени распознают переходные и установившиеся режимы и выдают экспертные заключения. Это позволяет проводить настройку двигателя без его остановки, что значительно снижает расход топлива.

### Программа Harna

Данная программа применяется для оценки и настройки характеристик управляющих аппаратов (рис. 5).

### Программа Alarm

Для аварийных, предаварийных и информационных сообщений используется программа **Alarm**. Она занимается приёмом сообщений от всех задач АИИС и выводит их на экран (рис. 6). Для каждого испытания ведётся журнал регистрации сообщений. Кроме этого, программа **Alarm** занимается постоянным мониторингом системы и в случае опасности незамедлительно активирует реакцию программной защиты.

### Программа Recorder

Для записи и воспроизведения используется программа **Recorder**. Программа имеет два режима записи. В нормальном режиме программа пишет параметры с частотой 10 Гц. При выходе двигателя на переходный режим запись ускоряется до 50 Гц. Для гарантированной записи была реализована запись в формате RDM с использованием транзакций. После

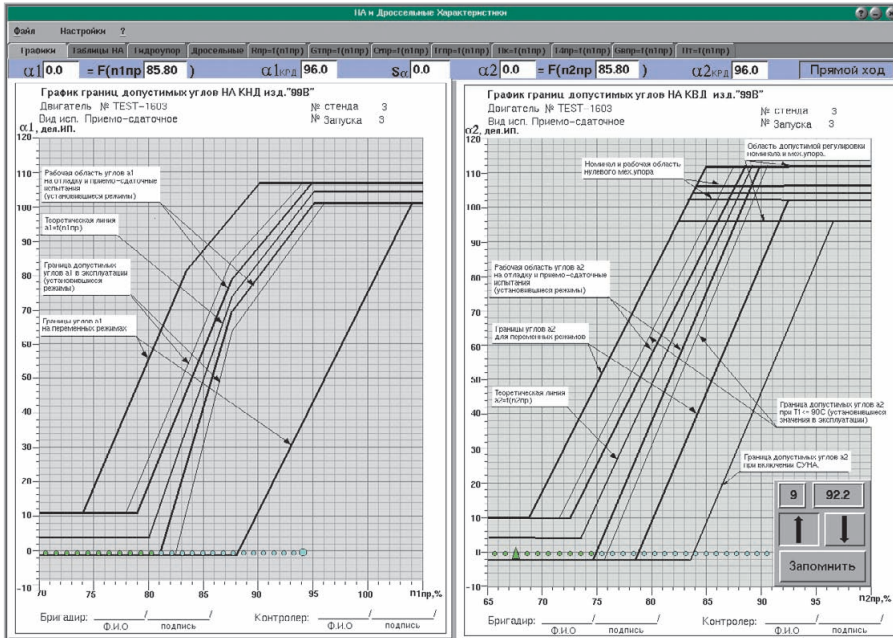


Рис. 5. Экран программы Napta

окончания испытания автоматически формируется запакованный текстовый файл записи. Кроме функции записи, **Recorder** позволяет воспроизвести любое записанное испытание в режиме реального времени либо в режиме быстрого просмотра. При воспроизведении драйверы работы с интерфейсными платами останавливаются, и программа **Recorder** начинает высылать записанные коды в **receiver**. Это позво-

ляет просматривать реакции наших программ во время испытания и проследить за действиями механика-испытателя.

### Программа SuperVision

После окончания предъявительских испытаний двигателя все данные по испытанию предоставляются заказчику. Для удобного просмотра этих данных в ОС Windows была создана про-

2	23-03-2012	10:20:13.2	Рассогласование ПУРС по ДБСКТ (1)(2)(3)	
3	23-03-2012	10:20:24.2	Рассогласование ПУРС по ДБСКТ (1)(2)(3)	
4	23-03-2012	10:20:25.8	Запись параметров включена	
5	23-03-2012	10:20:35.2	Рассогласование ПУРС по ДБСКТ (1)(2)(3)	

Рис. 6. Пример сообщений программы Alarm

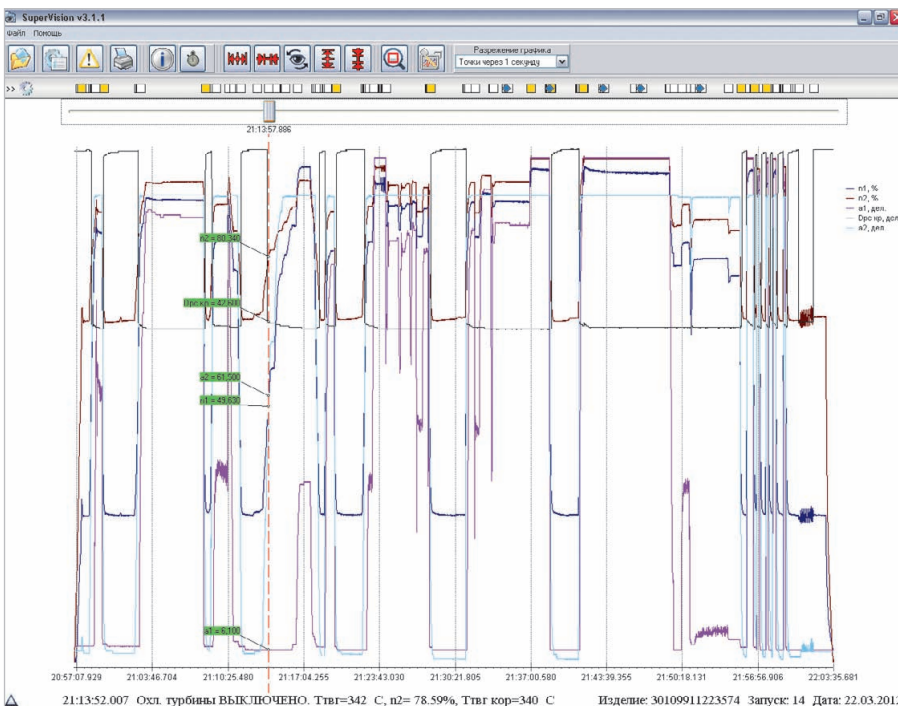


Рис. 7. Экран программы SuperVision

грамма **SuperVision**. Она позволяет просматривать значения параметров в графическом виде с параллельным выводом на текущий момент сообщений из журнала регистрации **Alarm** (рис. 7). Вывод сообщений подсвечивается разным цветом: белым – информационные, жёлтым – предупредительные, красным – аварийные. Программой поддерживается просмотр записей, сделанных АРМДК (автоматизированное рабочее место диагностического комплекса), программами **Recorder**, **Fast**, **Alarm**, **Wpoint**, а также протокола испытания.

### НОВЫЕ ПРОЕКТЫ

В новых проектах мы планируем использовать операционную систему QNX 6, которая является современной и широко используемой ОС реального времени для широкого круга ответственных применений.

Достоинства QNX 6, которые нас привлекают:

- более широкая поддержка аппаратных средств, в том числе и процессоров различной архитектуры;
- поддержка нитей в рамках одного процесса;
- более современная и удобная среда разработки и отладки прикладного ПО QNX Momentics IDE на базе платформы Eclipse (правда, среда разработки функционирует в ОС Windows, и требуется дополнительный компьютер, соединённый с целевой системой Ethernet-сетью);
- новые встроенные механизмы повышения надёжности функционирования прикладных систем, такие как NAM – мониторинг для восстановления работы приложения при сбое и Adaptive Partitioning – гарантированное распределение ресурсов между группами процессов.

Наверняка в QNX 6 есть ещё много новых достоинств, о которых мы пока можем только догадываться.

Также не является догмой описанная в данной статье структура ПО. Для обеспечения независимой работы особо ответственного контроллера реализован вариант работы на нём дополнительных сервера параметров и сервера данных испытаний.

Если у нас получится что-нибудь стоящее, мы об этом снова напишем.

E-mail: lhait1@yandex.ru