

## 6. Программируемые контроллеры, устройства/модули сопряжения с объектами и их компоненты (примеры)

### 6.1. ПТК на основе контроллера для распределенных открытых систем (КРОСС) и инструментальной программной системы (среды) для проектирования программного обеспечения контроллеров *ISaGRAF*

Контроллер КРОСС (ОАО “ЗЭиМ”, Чебоксары) предназначен для общепромышленного применения в составе автоматизированных систем управления техническими объектами и технологическими процессами, а также в качестве автономного средства для управления объектами малой и средней сложности (рис. 6.1). В нем используется операционная система реального времени, система технологического программирования ISaGRAF.

*Программное обеспечение (ПО) контроллера позволяет:* а) осуществить контроль, управление и тестирование каналов ввода-вывода автономно и с помощью компьютера; 2) снизить затраты на разработку и отладку программ пользователя за счет простоты и удобства программирования, их переноса и документирования.

Его программно-аппаратные средства обеспечивают обмен данными в реальном времени через интерфейс *Ethernet*.

Развитые системообразующие качества, *позволяют:* интегрировать в единую систему контроллеры различных производителей, выполненные в стандартах открытых систем; масштабировать системы; использовать единую технологию программирования контроллеров различных фирм и переносимость технологических программ пользователя.

В контроллере применены *интеллектуальные* УСО, имеющие встроенные «бортовые» микроконтроллеры, выполняющие независимо и асинхронно по отношению к центральному процессору (ЦП) различные функции по обработке сигналов и диагностике оборудования. Это *позволяет:*

- повысить производительность и уменьшить время цикла за счет сокращения нагрузки на ЦП по объему вычислений и интенсивности обменов данными с модулями УСО;
- повысить живучесть системы АСУ ТП за счет децентрализации и автономного выполнения функций; иметь возможность контроля, управления и тестирования модуля УСО в автономном режиме (без ЦП) с помощью компьютера через порт модуля с интерфейсом RS-232;
- получить масштабируемость подсистемы ВВ.

Модули УСО автономно, без участия ЦП, управляют в циклическом режиме процедурами ввода-вывода, АЦ- и ЦА-преобразований и др.

*Высокая надежность ПРК обеспечивается следующими решениями:*

- использование микромощной элементной базы ведущих зарубежных фирм;
- исполнение системных и технологических программ из *flash-памяти* компьютера;
- использование высоконадежного ПО, имеющего сотни тысяч инсталляций;
- аппаратная поддержка сетевых протоколов;
- резкое снижение числа межмодульных контактных соединений за счет использования последовательной *внутриконтроллерной магистрали SPI* (4 сигнальных провода на модуль);
- непрерывная внутримодульная диагностика, наличие сторожевых таймеров во всех интеллектуальных модулях; широкие возможности резервирования.

#### *Состав контроллера*

КРОСС имеет проектно-компоновемый состав, конкретный контроллер состоит из устройств, число и типы которых определяются заказом потребителя (рис. 6.1):

- центрального *блока* (ЦБ1),
- *блока* питания (ЛОК 4601-2R), а также *модулей* питания,
- *модулей* УСО,
- соединителей SPI,
- соединителей гибких и терминальных блоков.

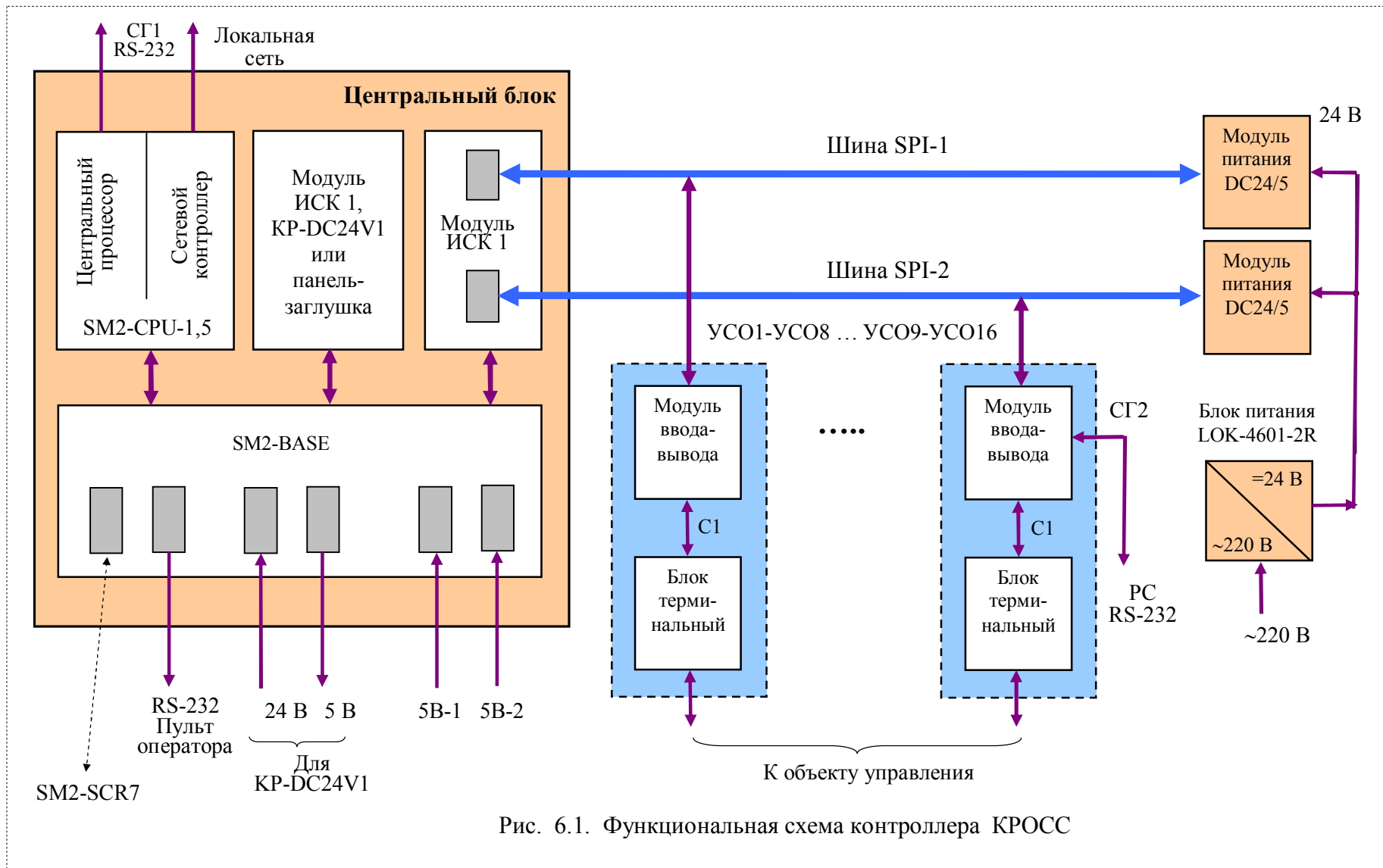


Рис. 6.1. Функциональная схема контроллера КРОСС

ЦБ1 состоит из базового монтажного блока SMART2-BASE и установленных на него модулей. Модуль процессора SM2-CPU-1,5 состоит из ЦП и *сетевого контроллера*. Всегда устанавливается модуль процессора, а также модуль ИСК1 в зависимости от заказа, или модуль питания KP-DC24V1.

Модули УСО содержат микроконтроллер. Обмен между процессорами выполняется по шинам SPI через модуль ИСК1. К каждой шине SPI можно подсоединять до 8 модулей УСО. В зависимости от их числа этих модулей ИСК1 может быть от 1 до 2 шт.

Модуль ИСК1 *соединяет и согласовывает модули* УСО с ЦП. Они взаимодействуют через 2 шины SPI. На каждой шине может быть до 8 модулей УСО. Общее число модулей УСО - до 31 по четырем шинам.

Для связи с внешними приборами по локальной сети *Ethernet* на модуль ЦП может устанавливаться submodule Ethernet SM2-ETH. Модуль ЦП является мастером шины SPI и предназначен для управления работой ПРК, организации обмена с внешними устройствами, а также взаимодействия пользователя с ПРК через компьютер и SCADA-систему.

*В общем случае ПРК может выполнять следующие функции:*

- измерение сигналов датчиков, фильтрацию, линеаризацию и преобразование принятых сигналов в цифровое нормализованное представление;
- прием дискретных сигналов;
- управление, регулирование, вычисление в соответствии с программой пользователя;
- формирование и вывод управляющих аналоговых и дискретных сигналов на ОУ;
- хранение программ и констант пользователя, а также служебных программ во *flash*-памяти, переменных процесса в статическом энергонезависимом ОЗУ;
- самодиагностика модулей и ПРК в целом;
- информационный обмен:
  - с другими ПРК и компьютерами через локальную сеть Ethernet;
  - с компьютером через последовательный интерфейс RS-232;
  - сопряжение ПРК с различными SCADA-системами;
  - отображение информации на экране компьютера, подключенного к ПРК через интерфейс RS-232 или через сеть Ethernet.

Каждый *модуль УСО* функционально построен одинаково и содержит процессор, схему приема/выдачи аналогового или дискретного сигнала.

Обмен между центральным блоком ЦБ1 и модулями УСО ведется по синхронному интерфейсу SPI. Базовый монтажный блок содержит микроконтроллер MC68HC705C8A.

*Модуль ЦП* поддерживает программное обеспечение ISaGRAF и операционную систему OS-9 и *выполняет функции:*

управление контроллером; администрирование submodule; оперативный мониторинг контроллера.

Совместно с submodule процессор представляет собой *интеллектуальное и коммуникационное* устройство управления.

*Submodule SM2-ETH* представляет собой полнодуплексный коммутируемый контроллер *Ethernet* с интерфейсом 10Base-T (10 Мбит/сек), подключается витой парой к внешним цепям через разъем "ETHERNET".

### *Программное обеспечение (ПО) контроллера*

ПО контроллера состоит из *резидентного* (*встроенного* в ПРК), программы эмулятора пульта настройки и сервисной программы "CrossTest" для проверки модулей УСО.

В состав резидентного ПО входят ОСРВ OS-9 и исполнительная среда системы технологического программирования ISaGRAF.

OS-9 – высокопроизводительная система, базирующаяся на микроядре с наращиваемой модульной архитектурой, поддерживает малые, средние и большие системы, обладает развитой иерархической системой ВВ, включая коммуникации по последовательным портам, дисковые устройства, сетевые протоколы *Ethernet*, TCP/IP, ISDN и др.

Благодаря уникальной архитектуре OS-9 может работать и в минимальной конфигурации, выполняясь непосредственно из ПЗУ и занимая менее 100 кбайт памяти.

Таким образом, КРОСС *содержит*: процессоры; память; средства коммуникации; устройства ввода данных от ДЧ и вывода управляющих воздействий на ИМ; средства индикации.

*Память* ПРК обеспечивает хранение ядра ОСРВ, необходимых утилит и прикладных программ управления объектом. В ней используются микросхемы *постоянной* и *оперативной* памяти.

*Средства коммуникации* ПРК реализуют дистанционную загрузку задач и оперативный обмен данными между ПРК, рабочими станциями и компьютерами верхних уровней.

*Средства интерфейса с оператором* выполняются с учетом типа системы. Обмен может осуществляться по витой паре, коаксиальному кабелю, волоконно-оптическому кабелю (особенно при работе в производственных условиях с высоким уровнем электромагнитных помех) или беспроводному каналу передачи данных.

ПРК предназначен для работы в режиме РВ в условиях промышленной среды и должен быть доступен для программирования неспециалистом в области информатики.

### ***Некоторые модули УСО, используемые в ПРК КРОСС и P130ISa:***

**АП1-8** - модуль *ввода* аналоговых сигналов с гальванической развязкой (*рис. 6.2*);

**АЮ1-8/4** - модуль *ввода/вывода* аналоговых сигналов, выполненный с групповой гальванической развязкой входов и выходов друг от друга (*рис. 6.3*).

#### **Плата ввода аналоговая АП1-8**

Входные аналоговые сигналы (0-5), (0-20), (4-20) мА, (0-10) В преобразуются сначала в напряжение (0-2) В, затем напряжение преобразуется в частоту. Количество импульсов за фиксированный период времени подсчитывается с помощью *двоичного счетчика*, который и является последним звеном АЦП.

Модуль АП1-8 содержит восемь гальванически развязанных друг от друга *ячеек* (DN).

Субмодуль питания предназначен для индивидуального питания ячеек аналоговых плат стабилизированным напряжением 5 В. Канал питания содержат автогенератор G1, трансформатор и выпрямители со стабилизатором (ST). Сглаживание пульсаций обеспечивают емкостные фильтры.

Каждая *ячейка* настраивается на один вид и диапазон входного сигнала. Входные аналоговые сигналы с помощью сменных резисторов делителей приводятся к одному диапазону (0-2) В, проходят через RC-фильтр для подавления помехи, преобразуется в частоту F, усиливаются по току и через гальваническую развязку (*оптопару*) поступают на разъем платы процессора. Выход оптопары выполнен по схеме с открытым коллектором.

#### **Плата ввода/вывода аналоговая АЮ1-8/4**

Функциональная схема платы приведена на *рис. 3*. Она имеет два независимых гальванически разделенных узла – аналогового ввода и аналогового вывода.

***Аналоговый ввод.*** Входные сигналы через разъем ввода поступают на 8 ячеек, где каждая ячейка настроена на один из диапазонов входных сигналов. Входные сигналы с помощью сменных элементов делителей приводятся к диапазону (0-2) В.

В состав ячейки входят RC - фильтр и диоды, защищающие входную цепь от перенапряжения и напряжения обратной полярности. После преобразования входной сигнал поступает на аналоговый мультиплексор, который выбирает тот или иной канал. С его выхода сигнал преобразуется в частоту F и усиливается по току. Оптопара обеспечивает гальваническую развязку входов от платы процессора и выходов аналоговой платы. Питание аналогового ввода производится от DC/DC преобразователя с гальванической развязкой.

***Аналоговый вывод.*** Используется интегральный четырехканальный ЦАП с последовательным вводом. Сигналы управления ЦАП поступают с платы процессора через оптопары. На выходе ЦАП включены преобразователи «напряжение-ток». ЦАП имеет внешний источник опорного напряжения. Питание выходных цепей производится от внешнего источника постоянного тока напряжением 24 В, питание ЦАП от стабилизатора 5 В.

Диапазон выходного сигнала определяется установкой измерительных сменных резисторов.

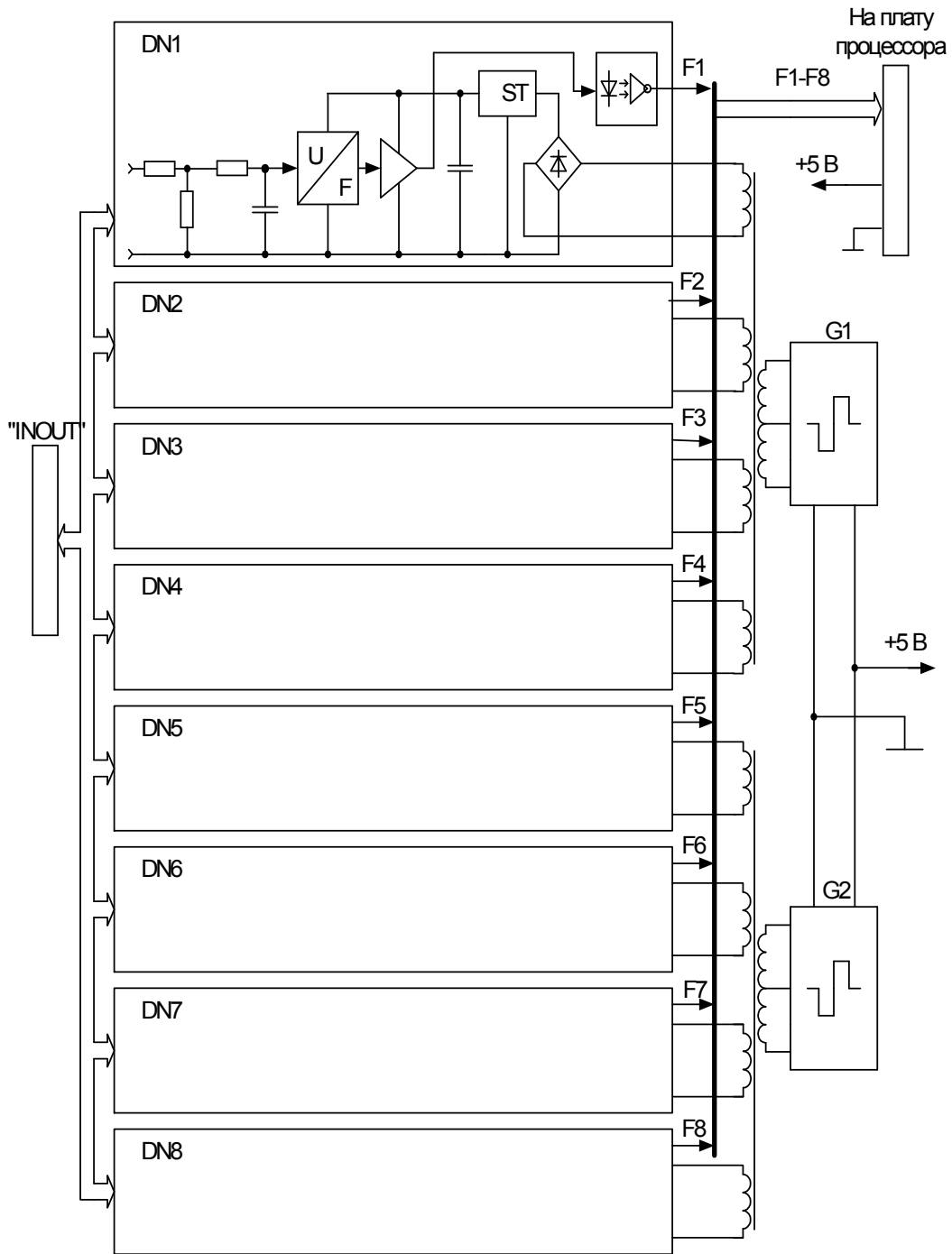


Рис. 6.2. Функциональная схема платы аналогового модуля AI1-8

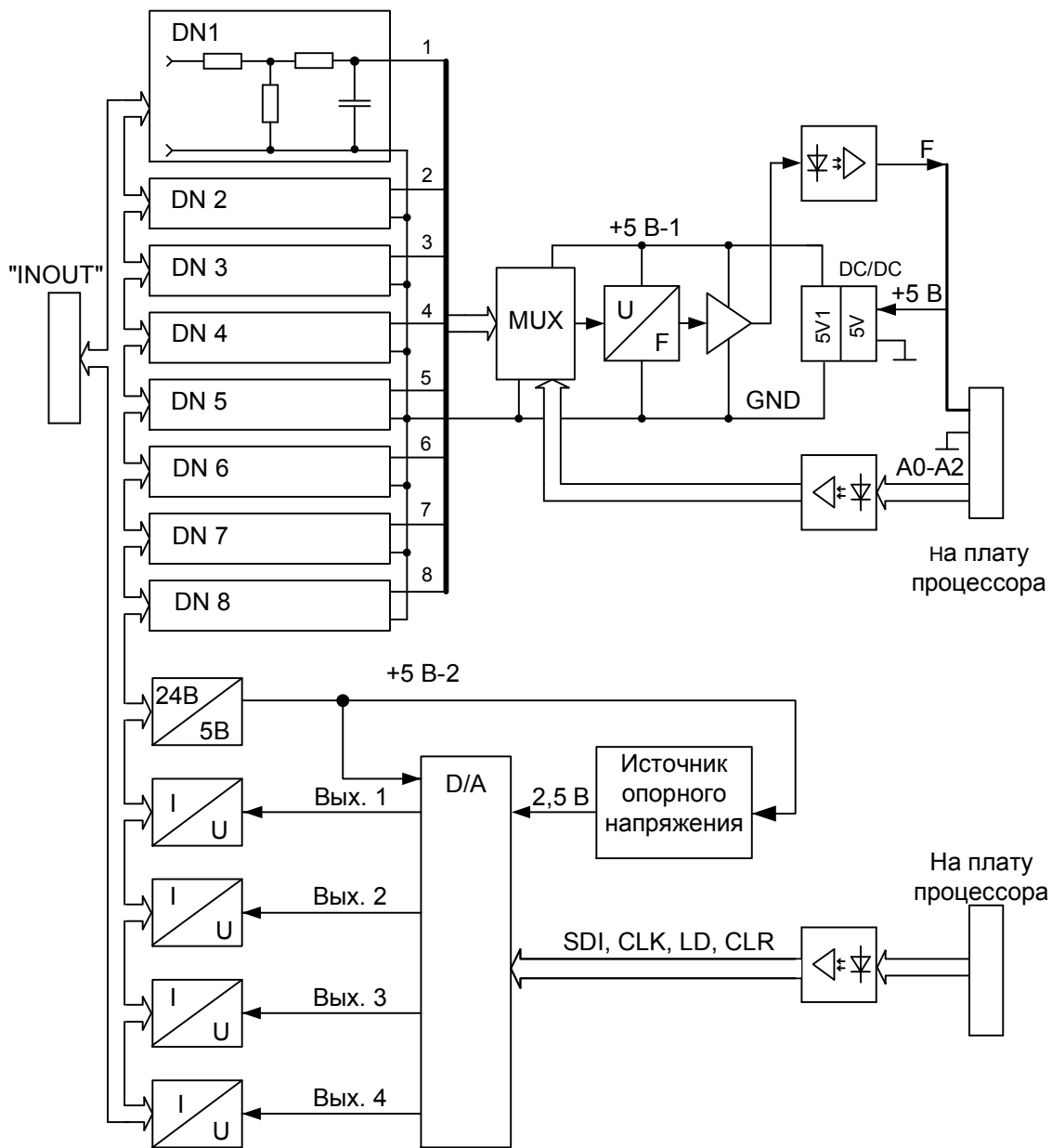


Рис. 6.3. Функциональная схема модуля AIO1-8/4

## Инструментальная программная система *ISaGRAF* для проектирования программного обеспечения контроллеров

*ISaGRAF* – это инструментальная программная система (среда) для проектирования программного обеспечения (ПО) контроллеров. В частности он является составной частью ПТК КРОСС и Р130ISa (ОАО “ЗЭИМ”, г. Чебоксары).

*ISaGRAF* является универсальным/интегрированным инструментом для автоматизации разработки качественного программного продукта не профессиональными программистами, а теми, кто занимается разработкой и обслуживанием сложных технических систем (в случае сложной задачи и приглашения *профи* очень большая вероятность попасть в зависимость от них).

Он состоит из двух частей (рис. 6.4): системы разработки *ISaGRAF Workbench* и системы исполнения *ISaGRAF Target*.

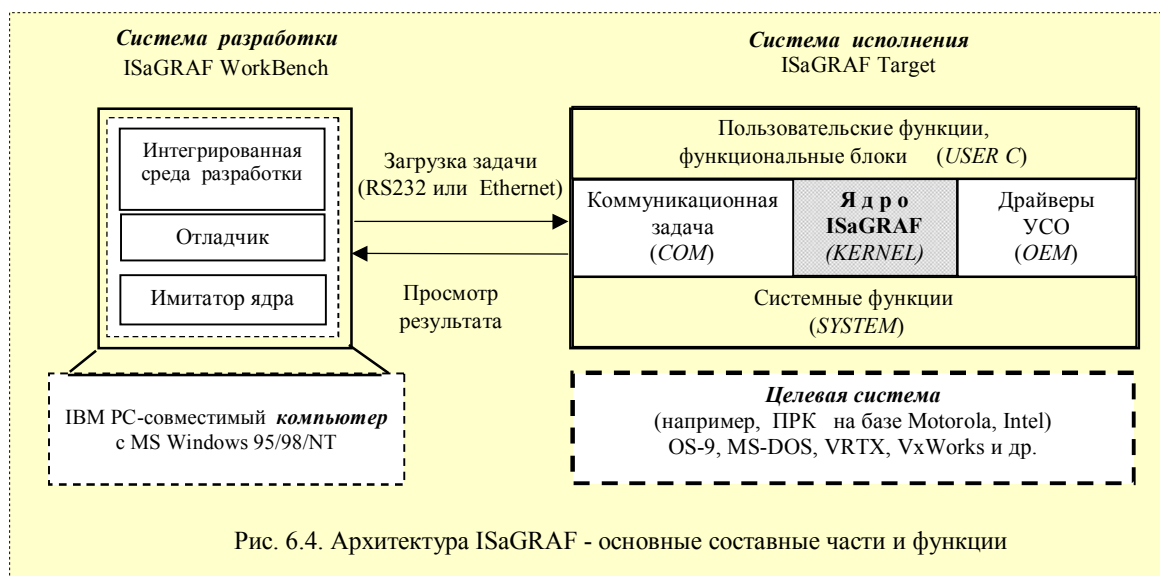


Рис. 6.4. Архитектура ISaGRAF - основные составные части и функции

Система разработки представляет собой набор *Windows-приложений*, интегрированных в единую инструментальную среду и работающих под управлением ОС Windows 95/98/NT.

Основу системы исполнения составляет набор программных модулей (для каждой целевой системы свой), выполняющих самостоятельные задачи под управлением так называемого ядра *ISaGRAF*.

В целом, ПО системы исполнения представляет собой прикладную программу реального времени, выполняемую на ППК (промышленной компьютерной системе или отдельной плате). Возможны две реализации такого ПО:

- *однозадачная* - где все функции выполняются в одной и той же программе;
- *многозадачная* - где для выполнения, например, функций связи предназначена отдельная задача.

Независимо от реализации, большинство функций системы исполнения являются общими (схематично они представлены на рис. 1).

### Основные возможности ISaGRAF:

- поддерживает все пять языков стандарта МЭК 61131, при этом позволяет смешивать программы и процедуры, написанные на разных языках, а также вставлять кодовые последовательности из одного языка в коды, написанные на другом языке;
- имеется возможность программной эмуляции ППК, которая удобна тем, что благодаря наличию многозадачной ОС можно совместить в одном месте контроллер, среду программирования и систему диспетчерского управления (например, SCADA-систему);
- имеет многофункциональный отладчик, позволяющий во время работы прикладной задачи просматривать состояние программного кода, переменных, программ и многое другое;
- поддерживает различные протоколы промышленных сетей;
- содержит набор драйверов для работы с различными модулями УСО, под управлением ППК различных фирм-производителей: PEP Modular Computers, Motorola Computer Group и др.;
- полное документирование этапов проектирования ПО.

Основной принцип, лежащий в основе системы исполнения – синхронизация. *ISaGRAF Target* – это *синхронная система*, управляемая таймером. Программируемый период срабатывания таймера называют *продолжительностью временного цикла*.

*Прикладная задача* работает строго по *временным циклам*, их длительность определяется разработчиком при компиляции задачи. Минимальная длительность циклов исполнения прикладной задачи определяется характеристиками аппаратно-программной платформы, на которой происходит исполнение задач: например, для OS-9 этот параметр равен 10 мс, для MS-DOS – не менее 55 мс.

Программный цикл начинается опросом всех сконфигурированных внешних каналов датчиков (например, каналы АЦП) и завершается обновлением всех выходных каналов (например, каналы ЦАП). Такая схема работы *приложения* гарантирует пользователю, что в рамках одного временного цикла он будет работать только с одной копией объектных данных типа INPUT/OUTPUT.

*Отладчик системы ISaGRAF* - предоставляет полный набор возможностей для получения качественного программного продукта. Использование средств программирования, поддерживающих открытые стандарты, дает очевидные преимущества:

- получение качественного программного продукта;
- совместимость на уровне исходных текстов;
- независимость от типа ОС и от конкретной персоны программиста;
- появление общего языка общения в среде разработчиков прикладного ПО;
- и самое важное - значительное сокращение времени разработки прикладного ПО и, как следствие, сокращение финансовых затрат на разработку проектов в целом.

## **6.2. ПТК на основе контроллера P-130Isa и инструментальной программной системы (среды) для проектирования программного обеспечения контроллеров ISaGRAF**

### **Ремиконт P-130 с блоком контроллера БК-1М/01**

Контроллер P-130 с БК-1М/01 построен на базе одноплатного PC-совместимого компьютера промышленного исполнения, имеет:

- а) открытую программно-аппаратную архитектуру;
- б) встроенную исполнительную систему ISaGRAF target kernel, позволяющую исполнять программы, написанные на технологических языках программирования, поддерживаемых системой ISaGRAF (стандарт МЭК публикация 61131-3); поэтому далее используется обозначение этого контроллера - P130Isa.

Встроенное ПО позволяет также выполнить проверку работоспособности и настройку контроллера.

P130Isa предназначен для автоматического регулирования и логического управления техническими объектами и технологическими процессами в различных отраслях промышленности.

Он эффективно решает как сравнительно простые, так и сложные задачи управления, благодаря малоканальности позволяет, с одной стороны, экономично управлять небольшим агрегатом и, с другой, обеспечить высокую живучесть крупных систем управления.

P130Isa позволяет вести локальное, каскадное, программное, супервизорное, многосвязное регулирование. Архитектура P130Isa обеспечивает возможность ручную или автоматически включать, отключать, переключать и реконфигурировать контуры регулирования, причем все эти операции выполняются безударно независимо от сложности структуры управления. В сочетании с обработкой аналоговых сигналов P130Isa позволяет выполнять также логические преобразования сигналов и вырабатывать не только аналоговые или импульсные, но и дискретные команды управления.

P130Isa содержит средства оперативного управления, расположенные на лицевой панели, позволяющие вручную изменять режимы работы, устанавливать задание, управлять ходом выполнения программы, вручную управлять исполнительными механизмами (ИМ), контролировать сигналы и индцировать ошибки. Стандартные аналоговые и дискретные датчики (ДЧ) и ИМ подключаются к контроллеру с помощью индивидуальных кабельных связей. Внутри P130Isa сигналы обрабатываются в цифровой форме.

P130Isa могут подключаться к *локальной* сети Ethernet.

P130Isa представляет собой комплекс технических средств. В его состав входит центральный микропроцессорный блок контроллера БК-1М/01 и ряд дополнительных блоков. Центральный блок преобразует аналоговую и дискретную информацию в цифровую форму, ведет обработку цифровой информации и вырабатывает управляющие воздействия. Дополнительные блоки используются для предварительного усиления сигналов терморпар и термометров сопротивления, формирования дискретных выходных сигналов на напряжение 220 В, организации внешних переключений и блокировок и т.п.

P130Isa является проектно-компоновемым изделием. Его состав и ряд параметров определяются потребителем и указываются в заказе. В него встроены развитые средства самодиагностики, сигнализации и идентификации неисправностей.



## Основные свойства

В P130ISa предусмотрено:

- до 8 независимых контуров регулирования, каждый из которых может быть локальным или каскадным, с аналоговым или импульсным выходом, с ручным, программным (в том числе многопрограммным) или супервизорным задатчиком;
- разнообразное сочетание (по заказу) аналоговых и дискретных входов-выходов (всего – 35 модификаций, см. табл. 4);
- ручная установка или автоподстройка любых коэффициентов в любых алгоритмах;
- безударное изменение режимов управления и безударное включение/отключение, переключение и реконфигурация контуров регулирования любой степени сложности;
- оперативное управление контурами регулирования с помощью 12 клавиш, 2<sup>x</sup> 4-разрядных цифровых индикаторов и набора светодиодов, позволяющих менять режимы, устанавливать задание, управлять исполнительными механизмами, контролировать сигналы, индцировать аварийные ситуации.

### 6.3 Режим реального времени и ограничения на применение ПРК

*Реальный масштаб времени.* Время реакции ПРК должно быть согласовано с динамикой ОУ (с реально протекающими физическими процессами). Запаздывание выдачи управляющих воздействий и замедление скорости счета могут привести к потере устойчивости системы.

К аналогичным последствиям может привести и увеличение интервала дискретизации информации, поступающей от датчиков в ПРК.

В общем случае *шаг дискретизации* по времени входного воздействия определяется на основе *теоремы Котельникова* (для случайных *процессов с ограниченным спектром*).

Для *детерминированных процессов* он выбирается часто исходя из условия *непревышения* регулируемой величиной *шага квантования по уровню за время, равное шагу дискретизации*.

*Суть теоремы Котельникова* - если процесс происходит с определенной частотой, то для дискретизации сигналов, характеризующих этот процесс, нужно использовать частоту, как минимум в 2 раза большую.

Время расчета выходных величин в ПРК принципиально представляет собой конечную величину.

Поэтому можно говорить только о допустимом интервале времени расчета воздействий и периоде его повторения, который является функцией скорости изменения состояния ОУ, т.е. его *инерционности*, и *скорости изменения* возмущающих воздействий.

*Рассмотрим пример.* В САР управляющее устройство является регулятором, реализующим закон регулирования, который устанавливает зависимость между управляющим воздействием  $u(t)$  и ошибкой (рассогласованием)  $\varepsilon(t)$ , т.е.  $u(t)=F(\varepsilon(t))$ , где  $u(t)$  – регулирующее воздействие,  $\varepsilon(t)$  - ошибка (рассогласование).

В непрерывных системах широко используются пропорционально-интегрально-дифференциальные (ПИД) регуляторы, которые описываются следующим уравнением:

$$u(t) = K \left[ \varepsilon(t) + \frac{1}{T_1} \int_0^t \varepsilon(\tau) d\tau + T_2 \frac{d\varepsilon(t)}{dt} \right] \quad (1)$$

где  $K$  – коэффициент передачи,  $T_1, T_2$  - постоянные интегрирования и дифференцирования соответственно.

Трансформируем его к виду, который может быть реализован на вычислительных средствах, в частности - ПРК. Для малых периодов дискретизации по времени  $T$  это уравнение может быть преобразовано в разностное, осуществляя замену *производной обратной конечной разностью* первого порядка, а *интеграла* - *конечной суммой*. Непрерывное интегрирование может быть представлено с помощью метода прямоугольников, трапеций и др. При использовании метода прямоугольников уравнение (1) в дискретной форме записывается в следующем виде:

$$u(k) = K \left\{ \varepsilon(k) + \frac{T}{T_1} \sum_{i=0}^{k-1} \varepsilon(i) + \frac{T_2}{T} [\varepsilon(k) - \varepsilon(k-1)] \right\} \quad (2)$$

Используем известное соотношение для конечной суммы

$$f_{\Sigma}(k) = \sum_{i=0}^{k-1} f(i)$$

В результате получим *нерекуррентный* алгоритм управления, в котором для вычисления суммы необходимо помнить все предыдущие значения сигнала ошибки  $\varepsilon(k)$  и каждый раз значение управляющего сигнала  $u(k)$  вычисляется заново.

Для реализации программ закона регулирования на вычислительных средствах, в частности в ПРК, более удобным является *рекуррентный* алгоритм, который характеризуется тем, что для вычисления текущего значения управляющего сигнала  $u(k)$  используется его предыдущее значение  $u(k-1)$  и некоторые коэффициенты.

Для этого необходимо из переменной  $u(k)$  вычесть переменную  $u(k-1)$ :

$$u(k-1) = K \left\{ \varepsilon(k-1) + \frac{T}{T_1} \sum_{i=0}^{k-2} \varepsilon(i) + \frac{T_2}{T} [\varepsilon(k-1) - \varepsilon(k-2)] \right\} \quad (3)$$

В результате вычитания из уравнения (2) уравнения (3) и соответствующих преобразований (группирование и приведение подобных) получим:

$$u(k) - u(k-1) = K \left\{ \varepsilon(k) \left( 1 + \frac{T_2}{T} \right) + \varepsilon(k-1) \left( \frac{T}{T_1} - 1 - \frac{T_2}{T} \right) + \frac{T_2}{T} \varepsilon(k-2) \right\}$$

$$u(k) = u(k-1) + K \left\{ \varepsilon(k) \left( 1 + \frac{T_2}{T} \right) + \varepsilon(k-1) \left( \frac{T}{T_1} - 1 - \frac{2 \cdot T_2}{T} \right) + \frac{T_2}{T} \varepsilon(k-2) \right\}$$

$$u(k) = u(k-1) + C_0 \cdot \varepsilon(k) + C_1 \cdot \varepsilon(k-1) + C_2 \cdot \varepsilon(k-2),$$

$$\text{где } C_0 = K \cdot \left( 1 + \frac{T_2}{T} \right), \quad C_1 = K \cdot \left( \frac{T}{T_1} - 1 - \frac{2 \cdot T_2}{T} \right), \quad C_2 = K \cdot \frac{T_2}{T}$$

Для малых периодов дискретизации  $T$  коэффициенты  $C_1, C_2, C_3$  вычисляются с использованием параметров  $K, T_1$  и  $T_2$  аналогового ПИД-регулятора.

Рассмотрим рекуррентное соотношение при аппроксимации непрерывного интеграла на основе метода *трапеций*.

$$u(k) = K \left\{ \varepsilon(k) + \frac{T}{T_1} \left( \frac{\varepsilon(0) - \varepsilon(k)}{2} + \sum_{i=1}^{k-1} \varepsilon(i) \right) + \frac{T_2}{T} [\varepsilon(k) - \varepsilon(k-1)] \right\}$$

$$u(k-1) = K \left\{ \varepsilon(k-1) + \frac{T}{T_1} \left( \frac{\varepsilon(0) - \varepsilon(k-1)}{2} + \sum_{i=1}^{k-2} \varepsilon(i) \right) + \frac{T_2}{T} [\varepsilon(k-1) - \varepsilon(k-2)] \right\}$$

$$u(k) - u(k-1) = K (\varepsilon(k) - \varepsilon(k-1)) + K \left[ \frac{T}{T_1} \left( \frac{\varepsilon(0) - \varepsilon(k)}{2} + \sum_{i=1}^{k-1} \varepsilon(i) \right) - \frac{T}{T_1} \left( \frac{\varepsilon(0) - \varepsilon(k-1)}{2} + \sum_{i=1}^{k-2} \varepsilon(i) \right) \right] +$$

$$+ K \left[ \frac{T_2}{T} [\varepsilon(k) - \varepsilon(k-1)] - \frac{T_2}{T} [\varepsilon(k-1) - \varepsilon(k-2)] \right] =$$

$$= K (\varepsilon(k) - \varepsilon(k-1)) + \frac{K \cdot T}{T_1} \left( \frac{-\varepsilon(k)}{2} - \frac{\varepsilon(k-1)}{2} \right) + \frac{K \cdot T}{T_1} \varepsilon(k-1) + \frac{K \cdot T_2}{T} \{ \varepsilon(k) + \varepsilon(k-2) \} =$$

$$= K \cdot \varepsilon(k) - K \cdot \varepsilon(k-1) - \frac{K \cdot T}{2T_1} \varepsilon(k) - \frac{K \cdot T}{2T_1} \varepsilon(k-1) + \frac{K \cdot T}{T_1} \varepsilon(k-1) + \frac{K \cdot T_2}{T} \varepsilon(k) + \frac{K \cdot T_2}{T} \varepsilon(k-2) =$$

$$= \varepsilon(k) \left[ K - \frac{K \cdot T}{2T_1} + \frac{K \cdot T_2}{T} \right] + \varepsilon(k-1) \left[ \frac{K \cdot T}{T_1} - \frac{K \cdot T}{2T_1} - K \right] + \frac{K \cdot T_2}{T} \varepsilon(k-2) =$$

$$= \varepsilon(k) \cdot C_1 + \varepsilon(k-1) \cdot C_2 + \varepsilon(k-2) \cdot C_3$$

$$\text{где } C_1 = K - \frac{K \cdot T}{2T_1} + \frac{K \cdot T_2}{T} = K \left( 1 - \frac{T}{2T_1} + \frac{T_2}{T} \right); \quad C_2 = K \left( -1 + \frac{T}{T_1} - \frac{T}{2T_1} \right); \quad C_3 = \frac{K \cdot T_2}{T}$$

$$u(k) = u(k-1) + C_0 \cdot \varepsilon(k) + C_1 \cdot \varepsilon(k-1) + C_2 \cdot \varepsilon(k-2)$$

Введем следующие обозначения (рис. 6.5):

$T_i$  - допустимый *период повторения интервала времени расчета*  $i$ -го управляющего воздействия (допустимое время цикла расчета);

$t_i$  - полное время расчета  $i$ -го управляющего воздействия (цикл расчета);

$t_{ввi}$  - время ввода в ПРК исходных величин для расчета  $i$ -го управляющего воздействия;

$t_{расчi}$  - время *расчета* величины  $i$ -го управляющего воздействия;

$t_{прi}$  - время на принятие решения о выдаче на ОУ  $i$ -го управляющего воздействия;

$t_{вывi}$  - время непосредственного вывода на ОУ вычисленного  $i$ -го воздействия.

Тогда можно утверждать, что система с ПРК в контуре управления будет работать в *реальном времени*, если выполняется условие  $T_i \geq t_i$  или после подстановки

$$t_i = t_{ввi} + t_{расчi} + t_{прi} + t_{вывi},$$

с учетом замены  $t_{ввi} + t_{вывi} = t_{обmi}$ .

Сумма величин  $t_{ввi}$  и  $t_{вывi}$  представляет собой время обмена  $t_{обmi}$  с ОУ в процессе вычисления  $i$ -го управляющего воздействия. В результате получим

$$t_i = t_{расчi} + t_{прi} + t_{обmi}.$$

Абсолютное значение величины  $T_i$  для разных ОУ в зависимости от назначения и типа систем может изменяться в значительных пределах.

*При выборе шага дискретизации по времени ( $T$ ) принимаются во внимание:*

- динамические свойства непрерывной части системы (постоянные времени);
- предъявляемые требования к качеству замкнутой системы;
- структура цифрового алгоритма управления (предполагаемого для реализации);
- спектральный состав воздействий;
- разрядность цифровой части (АЦП, ЦАП, микропроцессора, памяти – хотя при выборе разрядности, в свою очередь, необходимо учитывать шаг дискретизации по времени);
- шаг квантования по уровню; - стоимостные характеристики.

Часто выбор шага дискретизации по времени предшествует решению задачи синтеза алгоритма управления. На основании теоремы Котельникова можно определить верхнюю границу  $T$ :

$$f_d = 1/T - \text{частота дискретизации}; f_{\max} - \text{максимальная частота в спектре сигнала};$$

$$f_d \geq 2 \cdot f_{\max}, \quad 1/T \geq 2 \cdot f_{\max}, \quad T \leq 1 / (2 \cdot f_{\max}). \quad \omega_{\max} = 2 \cdot \pi \cdot f_{\max}, \quad \omega_d = 2 \cdot \pi \cdot f_d, \quad \text{откуда} \quad T \leq \pi / \omega_{\max}.$$

Иногда принимают  $\omega_{\max} \approx 3 \cdot \omega_c$ ,  $\omega_c$  - частота среза, определяется соотношением  $|W_{нч}(j\omega)| \leq 0,1$ .

На практике можно брать (с определенным запасом)  $\omega_d \approx (5 \dots 10) \cdot \omega_c$ .

Требуется, чтобы  $\Delta x$  не превышало единицы младшего разряда. Тогда для  $n$ -разрядного АЦП должно выполняться приведенное выше условие для максимальной частоты (ширины спектра) преобразуемого сигнала, т.е. величину погрешности можно определить, если задаться шириной спектра сигнала и временем преобразования (быстродействием) АЦП.

#### *Временной (рабочий) цикл ПРК*

Задачи управления требуют непрерывного циклического контроля, поэтому вычисления в ПРК всегда повторяются циклически. *Одна итерация*, включающая чтение (сканирование) сигнальных входов и цифровых каналов), выполнение программы управления и установку выходов и передачу по цифровым каналам, называется *временным* или *рабочим циклом* ПРК (рис. 6.6). Выполняемые действия зависят от значения входов ПРК, предыдущего состояния и определяются пользовательской программой.

При включении питания ПРК выполняет *самотестирование* и *настройку* аппаратных ресурсов, очистку ОЗУ данных, *контроль* целостности прикладной программы пользователя. Если прикладная программа сохранена в памяти, ПРК переходит к основной работе, которая состоит из постоянного повторения последовательности действий, входящих в *рабочий цикл*:

- 1) начало цикла;
- 2) чтение состояния входов;
- 3) выполнение программы пользователя;
- 4) запись состояния выходов;
- 5) обслуживание аппаратных ресурсов ПРК;
- 6) мониторинг системы исполнения;
- 7) контроль времени цикла;
- 8) переход на начало цикла.

В самом начале цикла ПРК производит физическое чтение входов. Считанные значения размещаются в области памяти входов. Т.е., создается полная одномоментная зеркальная копия значений входов.

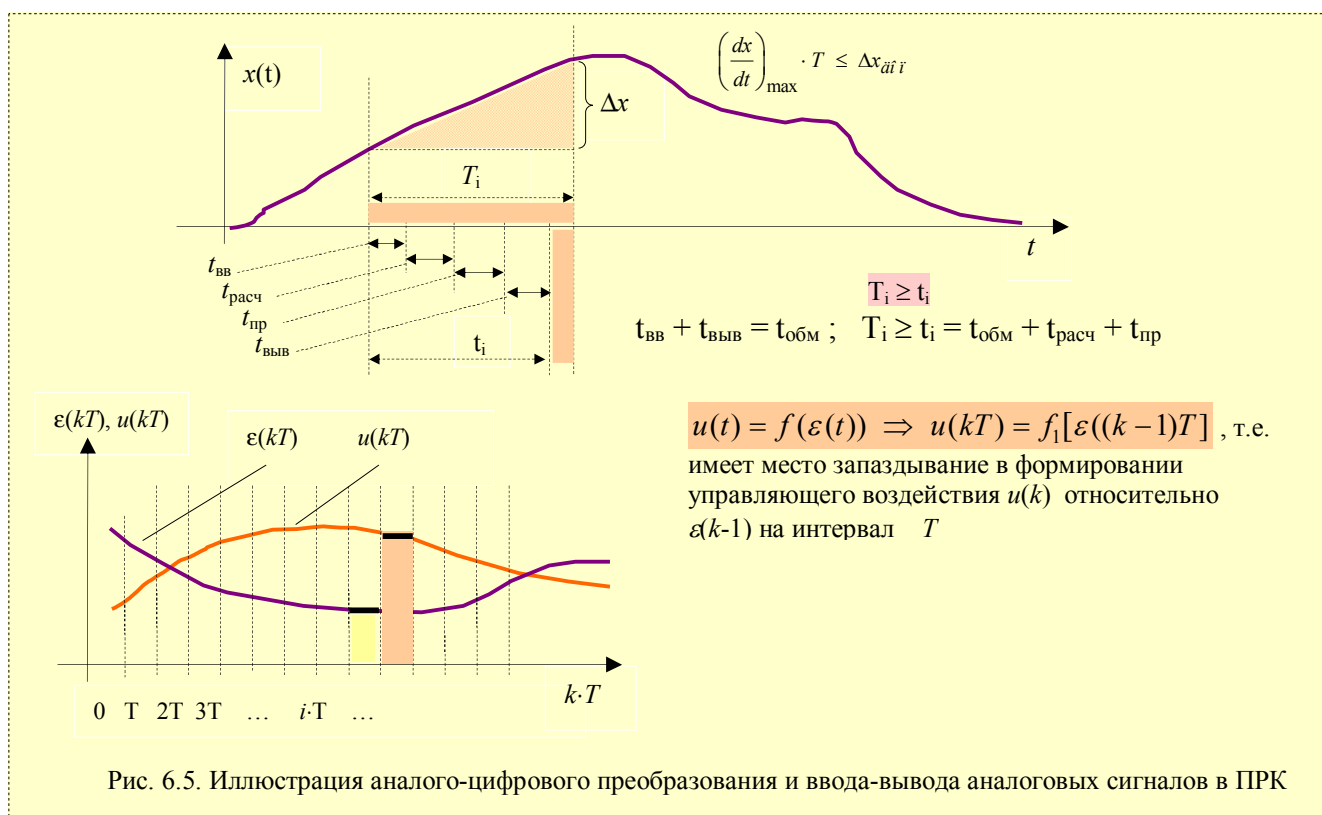


Рис. 6.5. Иллюстрация аналого-цифрового преобразования и ввода-вывода аналоговых сигналов в ПРК

Далее выполняется *пользовательская программа*, которая работает с копией значений входов и выходов, размещенной в ОЗУ. Если прикладная программа не загружена или остановлена, то данная фаза рабочего цикла, естественно, не выполняется. Отладчик системы программирования имеет доступ к образу входов-выходов, что позволяет управлять выходами вручную и проводить исследования работы датчиков.

После выполнения пользовательской программы физические выходы ПРК приводятся в соответствие с расчетными значениями.

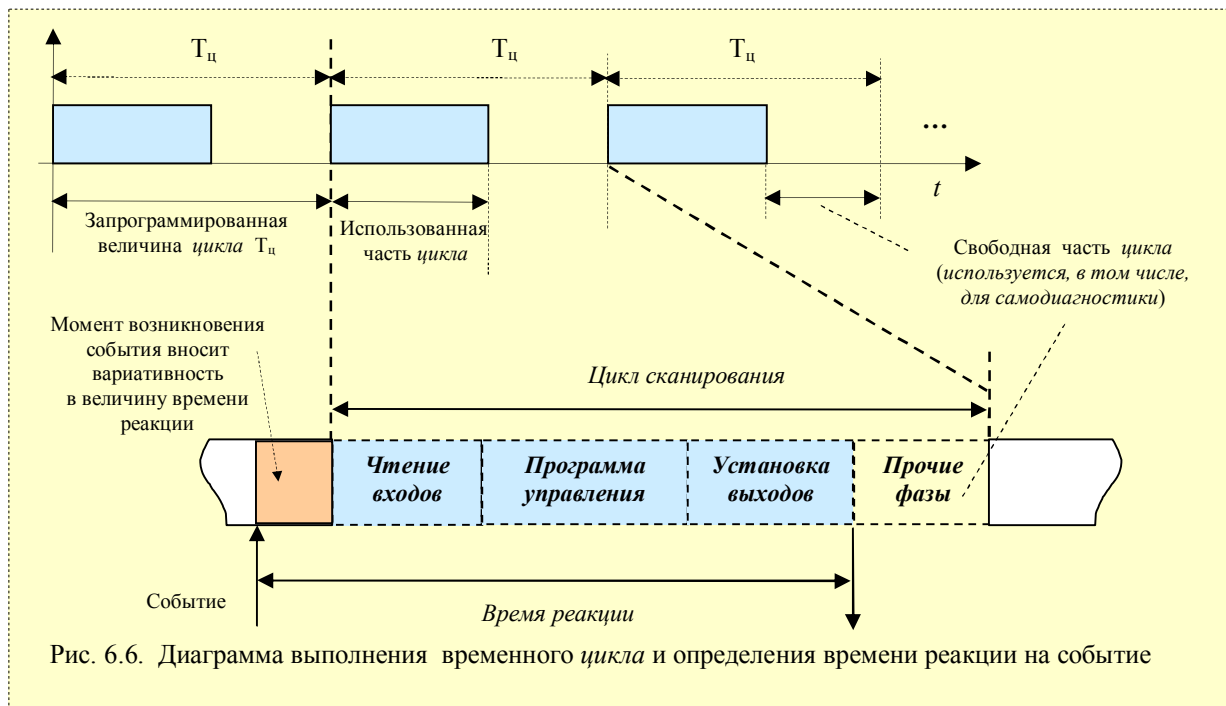


Рис. 6.6. Диаграмма выполнения временного цикла и определения времени реакции на событие

Типичные значения времени цикла ПРК находятся в пределах единиц или десятков миллисекунд.

*Обслуживание аппаратных ресурсов подразумевает:*

- 1) обеспечение работы системных таймеров;
- 2) часов реального времени;
- 3) оперативное самотестирование;
- 4) индикацию состояния и другие аппаратно-зависимые задачи.

*Монитор системы исполнения* включает большое число функций, необходимых при отладке программы и обеспечении взаимодействия с системой программирования, сервером данных и сетью.

*В функции системы исполнения обычно включаются:*

- загрузка исполняемого кода программы в ОЗУ и электрически репрограммируемую память,
- управление последовательностью выполнения задач,
- отображение процесса выполнения программ,
- пошаговое выполнение,
- обеспечение просмотра и редактирования значений переменных,
- фиксация и трассировка значений переменных,
- контроль времени цикла и др.

Пользовательская программа работает только с *мгновенной копией* входов. Таким образом, значения входов в процессе выполнения пользовательской программы не изменяются в пределах одного рабочего цикла. Это фундаментальный принцип построения ПРК *сканирующего типа*. Такой подход исключает неоднозначность алгоритма обработки данных в различных его ветвях. Кроме того, чтение копии значения входа из ОЗУ выполняется значительно быстрее, чем прямое чтение входа. Аппаратно чтение входа может быть связано с формированием определенных временных интервалов, передачей последовательности команд для конкретной микросхемы или даже запросом по сети.

Если заглянуть глубже, то нужно отметить, что не всегда работа по чтению входов полностью локализована в фазе чтения входов. Например, АЦП обычно требуют определенного времени с момента запуска до считывания измеренного значения. Часть работы системного ПО ПРК выполняет по прерываниям. Грамотно реализованная система исполнения нигде и никогда не использует пустые циклы ожидания готовности аппаратуры. Для прикладного программиста все эти детали не важны. Существенно только то, что значения входов обновляются автоматически исключительно в начале каждого рабочего цикла.

Поскольку *время цикла* существенно влияет на результат, например при автоматическом регулировании, то для устранения этой проблемы в развитых ПРК предусмотрен контроль времени цикла. Если от-

дельные ветви управляющей программы выполняются слишком быстро, в рабочий цикл добавляется искусственная задержка. Если контроль времени цикла не предусмотрен, подобные задачи приходится решать исключительно по таймерам.

*Время реакции* – это время с момента изменения состояния системы до момента выработки соответствующей реакции. Очевидно, что для ПРК время реакции зависит от распределения моментов возникновения события и начала фазы чтения входов. Если изменение значений входов произошло непосредственно перед фазой чтения входов, то время реакции будет наименьшим и равным времени сканирования.

Худший случай, когда изменение значений входов происходит сразу после фазы чтения входов. Тогда время реакции будет наибольшим, равным удвоенному времени сканирования минус время одного чтения входов. Иными словами, время реакции ПРК не превышает удвоенного времени сканирования.

Помимо времени реакции ПРК, существенное значение имеет время реакции датчиков и исполнительных механизмов, которое также необходимо учитывать при оценке общего времени реакции системы.

*Контроль времени рабочего цикла.* Правильно составленная пользовательская программа не должна содержать бесконечных циклов. В противном случае управление системе исполнения не будет передано, и, соответственно, нормальное функционирование ПРК будет нарушено. Для преодоления данной проблемы служит *контроль времени цикла*. Он осуществляется при поддержке аппаратно реализованного “*сторожевого таймера*”. Если фаза пользовательского кода выполняется дольше установленного порога, то ее работа будет прервана.

Таким образом, достигается предсказуемое поведение ПРК при ошибках в программе и при “*зависании*” по причине аппаратных сбоев. Обслуживание *сторожевого таймера* выполняется в рабочем цикле ПРК. Выполнять эту операцию по прерыванию нельзя, т.к. при “*зависании*” процессора система прерываний достаточно часто продолжает исправно работать.

В системах реального времени помимо правильности решения определяющую роль играет *время реакции*. Логически верное решение, полученное с задержкой более допустимой, не является приемлемым.

Принято различать системы *жесткого* и *мягкого* реального времени (РВ). В системах *жесткого* РВ существует явно выраженный временной порог. При его превышении наступают необратимые катастрофические последствия.

В системах *мягкого* РВ характеристики системы ухудшаются с увеличением времени управляющей реакции. Система может работать плохо или еще хуже, но ничего катастрофического при этом не происходит.

Абсолютное большинство ПРК работают по методу *периодического опроса* входных данных (*сканирования*). ПРК опрашивает входы, выполняет пользовательскую программу и устанавливает необходимые значения выходов.

Специфика применения ПРК обуславливает необходимость одновременного решения нескольких задач. Прикладная программа может быть реализована в виде множества логически независимых задач, которые должны работать одновременно.

На самом деле ПРК имеет обычно один процессор и выполняет несколько задач *псевдопараллельно*, последовательными порциями. Время реакции на событие оказывается зависящим от числа одновременно обрабатываемых событий.

Рассчитать минимальное и максимальное значения времени реакции, конечно, можно, но добавление новых задач или увеличение объема программы приведет к увеличению времени реакции. Такая модель больше подходит для систем *мягкого* РВ.

Современные ПРК имеют типовое значение времени рабочего цикла, измеряемое единицами миллсекунд и менее. Т.к. время реакции большинства ИМ значительно выше, с реальными ограничениями возможности использования ПРК по времени приходится сталкиваться редко.

В некоторых случаях ограничением служит не время реакции на событие, а обязательность его фиксации, например работа с датчиками, формирующими импульсы малой длительности. Это ограничение преодолевается специальной конструкцией входов. Так, счетный вход позволяет фиксировать и подсчитывать импульсы с периодом во много раз меньшим времени рабочего цикла ПРК. Специализированные интеллектуальные модули (КРОСС) в составе ПРК позволяют автономно обрабатывать заданные функции, например модули управления сервоприводом.

## Особенности интеграции ПРК в систему управления

Модули ввода-вывода могут применяться в трех основных конфигурациях: *локальный*, *удаленный* и *распределенный* ввод-вывод. В зависимости от конфигурации ввода-вывода и осуществляется *интеграция (объединение) ПРК в систему управления*.

*Локальный* (местный) ввод-вывод – такая конфигурация в рамках *одной* стандартной монтажной панели (на которой находится также и процессорный модуль) может содержать от нескольких штук до нескольких десятков модулей ввода-вывода наряду с модулями процессора и источниками питания.

*Удаленный* ввод-вывод – если требуется большое число линий ввода-вывода для подключения к удаленным узлам (размещенным на расстоянии до нескольких км в зависимости от типа соединительного кабеля), то конфигурируется система удаленного ввода-вывода. Связь с удаленными модулями осуществляется через специальные процессоры.

*Распределенный* ввод-вывод необходим для систем, в которых имеются пространственно распределенных небольшие узлы.

Для решения задач интеграции используются *сетевые технологии*. Предполагается наличие следующих компонентов:

- набор ПРК;
- несколько дисплейных рабочих станций операторов;
- системная (промышленная) сеть, соединяющая ПРК между собой и ПРК с рабочими станциями.

ПРК каждого сетевого комплекса, как правило, могут иметь ряд модификаций, отличающихся друг от друга быстродействием, объемом памяти, возможностями по резервированию, способностью работать в разных условиях окружающей среды, числом каналов ввода-вывода. Это облегчает использование сетевого комплекса для разнообразных объектов, т.к. позволяет наиболее точно подобрать ПРК под отдельные элементы ОУ и разные функции контроля и управления.

В качестве дисплейных рабочих станций (*нультов оператора*) почти всегда используются компьютеры в обычном или промышленном исполнении.

*Промышленная сеть* может иметь различную структуру (*топологию*): шину, кольцо, звезду. Сети часто подразделяются на сегменты, связанные между собой повторителями и маршрутизаторами.

К передаче сообщений предъявляются жесткие требования: они гарантированно должны доставляться адресату, а для сообщений высшего приоритета, например, предупреждающих об авариях, также следует обеспечить указанный срок передачи сообщений.

Промышленные сети с характерной структурой и различными/особыми физическими каналами связи (радиоканалы, выделенные телефонные линии, оптоволоконные кабели) позволяют интегрировать узлы объекта, отстоящие друг от друга на многие десятки километров, в единую систему.

*Интеллектуальные приборы* - термин «интеллектуальные» для первичных устройств – это те устройства, в которые встроен микропроцессор (МП), что добавляет новые функциональные возможности, которых не было в аналогичных устройствах без МП.

Например, интеллектуальный датчик может давать более точные показания благодаря применению вычислений, обеспечивающих компенсацию нелинейности чувствительного элемента или температурной зависимости.

Такой датчик имеет возможность работать с разными типами чувствительных элементов, а также объединять одно или несколько измерений в одно новое измерение (например, объемный расход и температуру в весовой расход). И наконец, интеллектуальный датчик позволяет производить настройку на другой диапазон измерений или полуавтоматическую калибровку, а также осуществлять функции внутренней самодиагностики, что упрощает техническое обслуживание.

*Интерфейс человек-машина* - еще 10-15 лет назад диспетчерский пульт управления представлял собой табло с множеством кнопок и световых индикаторов. В настоящее время подобные пульта применяются только в очень простых случаях, когда можно обойтись несколькими кнопками и индикаторами. В более «серьезных» системах применяются персональные компьютеры (ПК). Появился целый класс ПО, реализующего *интерфейс человек-машина*. Широко используется также *мультимедиа-технология*, позволяющая на компьютере объединять в единый комплекс информацию различного характера (графическую, звуковую, текстовую, видео) и управлять ею в режиме диалога.

SCADA-системы - системы сбора данных и оперативного/диспетчерского управления - выполняются с обязательным применением средств мультимедиа. Помимо реального отображения процесса производства, хорошие SCADA-системы позволяют накапливать полученные данные, проводят их хранение и анализ, определяют критические ситуации и производят оповещение персонала по каналам телефонной- и радиосети, позволяют создавать сценарии управления.

Сетевые технологии (промышленные сети). Второй часто возникающей задачей является интеграция нескольких ПРК с целью синхронизации их работы. Здесь появляются сети, обладающие рядом специфических требований. В целом это требования, аналогичные требованиям к ПРК:

- режим реального времени (РВ);
- надежность в условиях промышленной среды;
- ремонтпригодность; - простота программирования.

Такой класс сетей получил название *промышленных сетей* (Fieldbus). Существует масса фирменных реализаций и достаточно много стандартов таких сетей (Bitbus, Modbus, Profibus, CANopen, DeviceNet и др.), позволяющих интегрировать аппаратуру различных фирм, но ни один из них нельзя признать доминирующим.

Распределенные системы управления. Благодаря продуктивному развитию средств сетевой интеграции появилась возможность создания *распределенных систем управления*, в которых каждый ПРК решает локальную задачу. Задача синхронизации управления выполняется компьютерами среднего звена АСУ. Распределенные системы выигрывают по надежности, гибкости монтажа и простоте обслуживания.

Доступность программирования. Одним из основных требований, предъявляемых к ПРК, было и остается возможность его эксплуатации существующим техническим персоналом и возможность быстрой замены старого оборудования. Поэтому нужны простые и наглядные языки, позволяющие излагать задачу в близких к применяемым технологиям категориях.

#### 6.4. Примеры применения преобработки и преобразования аналоговых сигналов (рис. 6.7)

**Пример 1. Преобразование аналогового сигнала** (выход датчика, например измеряющего температуру в диапазоне от  $-50$  до  $+150$  °С) при вводе его в ПРК.

*Введем следующие обозначения:*

$In(t)$  – входное напряжение (аналоговый сигнал), поступающее с датчика ( $0 \dots 10$  В);

$U_{\text{COD}}(kT)$  - код сигнала, измеренного АЦП;

$U_1(kT)$  – цифровой эквивалент значения физической величины (например, температуры в градусах Цельсия °С, или по Фаренгейту);

$U_2(kT)$  – значение физической величины, выраженное в процентах;

$y(kT)$  – регулирующее воздействие, выраженное в процентах;

$y_1(kT)$  – цифровой эквивалент регулирующего воздействия;

$Out(t)$  – аналоговое напряжение, подаваемое на исполнительный механизм;

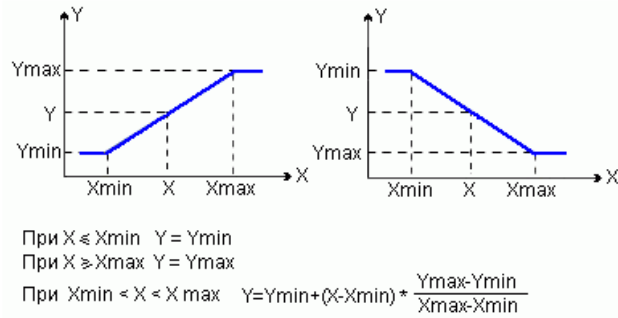
ФБ-1...ФБ-5 – функциональные блоки/алгоритмы (ФБ) из библиотеки инструментальной программной системы для конкретного контроллера.

ФБ-1 – преобразует код АЦП в цифровой эквивалент физической величины. *Например* в библиотеке КОНГРАФа есть функциональный блок (ФБ) «ТЕРМ 10К - Термистор 10 кОм», который предназначен для нормализации измерений датчика температуры (термистора 10 кОм), т.е. преобразования измеренного сопротивления датчика в значение температуры.

*Примечание: Термистор* (терморезистор) - сопротивление, изменяющее свою величину под действием температуры. Это теплоэлектрический полупроводниковый прибор, в котором используется зависимость электрического сопротивления полупроводника от температуры. Применяется для регистрации, измерения температуры в системах теплового контроля, в измерителях мощности и других устройствах.

ФБ-2 – преобразует значения физической величины в проценты. *Например* в библиотеке КОНГРАФа есть ФБ «ФИЗ ВЕЛ ОГ - Преобразование в физические единицы по двум точкам с ограничением», который можно использовать для этих целей, т.к. он выполняет линейное преобразование сигнала от датчика в физические единицы с ограничением.





X – сигнал датчика (мА, мВ или В);

Y – значение сигнала датчика, но уже в выбранных физических единицах.

ФБ-3 – преобразует входной сигнал, заданный в процентах, в соответствии с законом регулирования. *Например* в библиотеке КОНГРАФа есть ФБ «ПИД АНЛГ Р - ПИД-Регулятор аналоговый с ручным управлением» и ряд других.

ФБ-4 – ФБ, не требующий представления входных величин в процентах;

ФБ-5 – преобразует входные величины, заданные в процентах, в цифровой эквивалент регулирующего воздействия. *Например* в библиотеке КОНГРАФа есть ФБ «ФИЗ ВЕЛ ОГ - Преобразование в физические единицы по двум точкам с ограничением», который можно использовать для этих целей, т.к. он выполняет линейное преобразование сигнала от датчика в физические единицы с ограничением.

**В примере 1 считается заданным:**

- при изменении температуры на  $+1^{\circ}\text{C}$  напряжение на выходе датчика изменяется на  $+22,5$  мВ;
- температуре  $0^{\circ}\text{C}$  соответствует напряжение  $1,375$  В;
- диапазон измеряемой датчиком температуры от  $-50^{\circ}\text{C}$  до  $+150^{\circ}\text{C}$ ;
- 12-разрядные АЦП и ЦАП.

*Преобразование* (в ФБ-1) значения физического сигнала (напряжения на выходе датчика), которое представлено в кодах АЦП  $U_{\text{COD}}(kT)$ , в цифровой эквивалент физической величины - температуры, измеряемой датчиком,  $U_1(kT)$ , *можно представить следующим соотношением:*

$$U_1(kT) = \frac{\left( (U_{\text{COD}}(kT) - 2048) \cdot \left( \frac{V_{\max}}{K \cdot U_{\text{COD}}^{\max}} \right) - 1,375 \right)}{\Delta V}$$

При диапазоне положительных температур и коэффициенте усиления 1 формула примет вид:

$$U_1(kT) = \frac{U_{\text{COD}}(kT) \cdot \frac{V_{\max}}{U_{\text{COD}}^{\max}} - 1,375}{\Delta V} = \frac{\left( U_{\text{COD}}(kT) \cdot \frac{10 \text{ В}}{4095} - 1,375 \text{ В} \right) \text{ В}}{0,0225 \text{ В}/^{\circ}\text{C}} = \frac{U_{\text{COD}}(kT) \cdot \frac{10}{4095} - 1,375}{0,0225} \left[ ^{\circ}\text{C} \right]$$

*Здесь обозначено:*

2048 – поправка для смещения диапазона АЦП в пределы от -2048 до +2047 (учет отрицательных температур);

$V_{\max}$  – максимальное значение диапазона входного сигнала АЦП (0...10 В);

$K$  – значения коэффициента усиления входных цепей АЦП: 1, 10, 100;

$U_{\text{COD}}^{\max}$  – максимальное значение кода АЦП (4095);

1,375 – напряжение на датчике при  $0^{\circ}\text{C}$ ;

$\Delta V$  – приращение напряжения на датчике на  $1^{\circ}\text{C}$  (0,0225 В/ $^{\circ}\text{C}$ ).

После расчета температуры по приведенной формуле, значение переменной ограничивается диапазоном от -50 до +150, т.е. областью достоверных показаний датчика температуры.

Еще раз отметим те преобразования, которые должны быть реализованы в УСО. Контролируемый параметр – уровень напряжения и т.п. – с помощью соответствующего ДЧ преобразуется в электрический сигнал. Если сигнал на выходе ДЧ недостаточен для последующей обработки, то он усиливается до необходимого уровня. Далее с помощью фильтра (активного) из сигнала удаляются нежелательные низкочастотные и высокочастотные составляющие.

Отфильтрованный сигнал поступает через аналоговый мультиплексор на АЦП, который преобразует уровень напряжения на входе в соответствующую цифровую величину. Если входная величина некоторого ФБ (в нашем примере – это ФБ-3) должна быть представлена в процентах, то необходимо выполнить следующее линейное преобразование (в нашем примере оно выполняется в ФБ-2, а при выводе из ПРК – обратное преобразование осуществляется в ФБ-5). Формула такого преобразования примет вид (см. выше

график в описании ФБ-2):

$$y_1(t) = y_{1min} + [y(t) - y_{min}] \cdot \frac{y_{1max} - y_{1min}}{y_{max} - y_{min}} \quad [\%]$$

**Пример 2. Преобразование для получения соответствия** между регулирующим воздействием, представленным в процентах, и аналоговой величиной воздействия, приложенного к ИМ. Это преобразование осуществляется в ФБ-5, т.е. переменная  $y(kT)$ , заданная в процентах, преобразуется в коды ЦАП (переменная  $y_1(kT)$ ). Далее эта переменная преобразуется в аналоговое напряжение и подается на ИМ).

Например, сигнал управления  $y(kT)$  выражает угловое положение привода в процентах, а выходная переменная  $Out(t)$  – значения аналогового напряжения, подаваемого на ИМ.

Так как используется 12-разрядный ЦАП, то максимальное значение кода, равное  $2^{12}-1$ , обозначим как  $MAX$ . Будем считать это значение за 100%, а значит на 1% приходится  $MAX/100$ .

Для ИМ *Belimo* функции ФБ-1 выполняет ФБ «НАПРЯЖЕНИЕ – Напряжение (0-10В)» (имеющийся в КОНГРАФе), он преобразует напряжение, поступающее от датчика (0-10В) на вход АЦП через конфигурацию, в выходное значение напряжения датчика.

Так как управляющее напряжение, подаваемое на *Belimo* это напряжение постоянного тока, то можно оперировать целыми без знака, тогда  $MAX=2^{12}-1=4095$ . Значит на 1% приходится  $4095/100=4,095$ .

ФБ-3, представляющий собой реализацию ПИД-закона регулирования (например, ПИД АНЛГ Р для ПТК КОНТАР), ограничивает выходной сигнал  $y(kT)$  диапазоном 0 - 100%. Тогда значение переменной, выраженное в кодах ЦАП, будет эквивалентно значению  $y(kT)$ , выраженному в процентах, т.е.

$$y_1(kT) = y(kT) \cdot 4,095. \quad \text{Например,} \quad y(kT) = 45\%, \quad \text{тогда} \quad y_1(kT) = 45 \cdot 4,095 = 184,275.$$

*Программируемый контроллер*

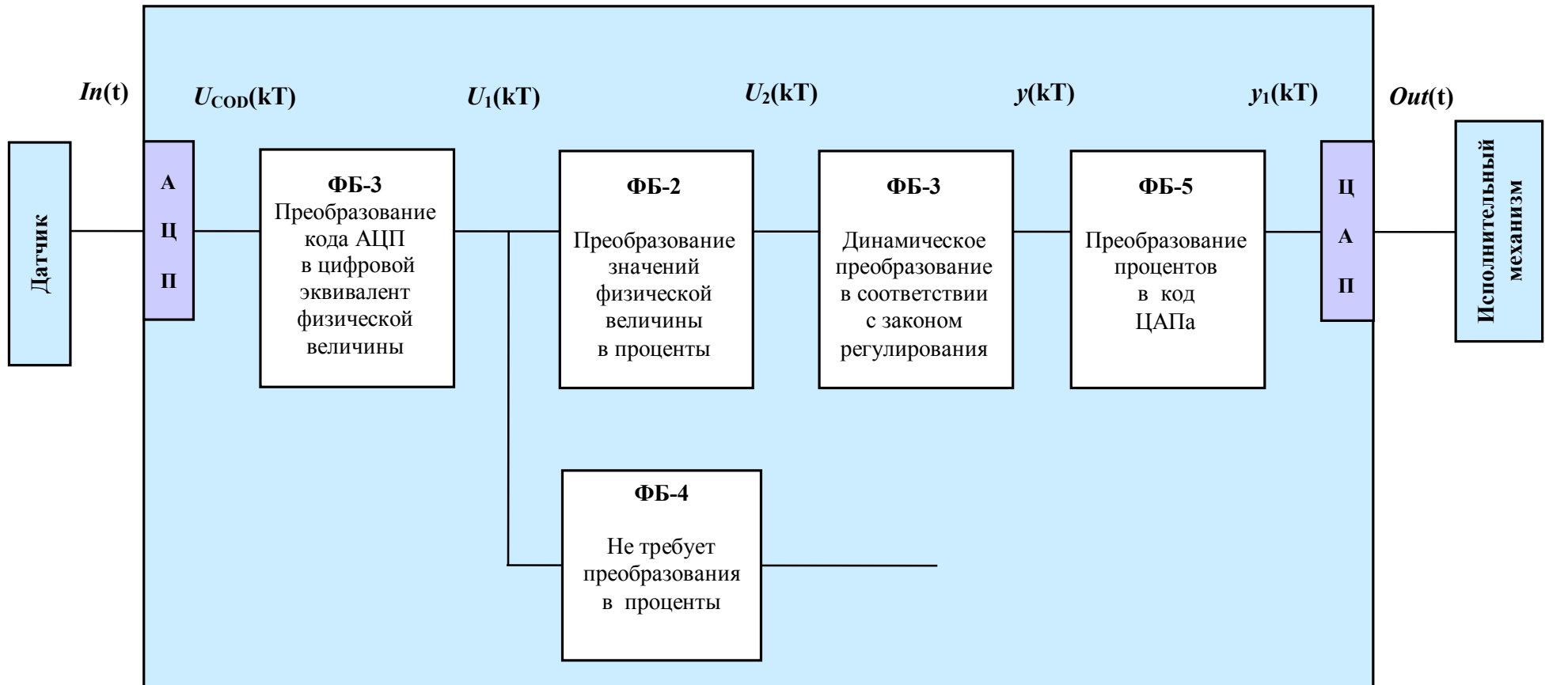


Рис. 6.7. Схема преобразований аналогового сигнала при вводе в контроллер и выводе из него цифрового сигнала на аналоговый исполнительный механизм: ФБ – функциональный блок (элемент библиотеки алгоритмов инструментальной программной системы); АЦП – аналого-цифровой преобразователь; ЦАП – цифро-аналоговый преобразователь