

Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра электронной техники и технологии

## **ПРОГРАММНО-УПРАВЛЯЕМЫЕ МИКРОКОНТРОЛЛЕРНЫЕ УСТРОЙСТВА**

*Рекомендовано УМО по образованию в области информатики и радиоэлектроники в качестве пособия по практическим занятиям для специальностей 1-36 04 01 «Программно-управляемые электронно-оптические системы», 1-39 02 02 «Проектирование и производство программно-управляемых электронных средств», 1-39 02 03 «Медицинская электроника»*

Минск БГУИР 2016

УДК 004.31-022.53(076)

ББК 32.973.26-04я73

П78

Авторы:

М. В. Давыдов, А. В. Фролов, Н. С. Давыдова, М. М. Меженная,  
А. М. Воробей, П. В. Камлач, Г. П. Будько

Рецензенты:

кафедра спортивной инженерии Белорусского национального  
технического университета (протокол №10 от 20.04.2015);

доцент кафедры конструирования и производства приборов  
Белорусского национального технического университета, кандидат  
технических наук, доцент Е. Г. Зайцева

**Программно-управляемые** микроконтроллерные устройства :  
П78 пособие / М. В. Давыдов [и др.]. – Минск : БГУИР, 2016. – 72 с. : ил.  
ISBN 978-985-543-200-6.

Рассматриваются основы работы с программным обеспечением AL-PCS/WIN-E, программирование функциональных блок-схем (FBD) контроллеров серии ALPHA 2, особенности мониторинга в схематическом представлении, режим моделирования контроллеров серии ALPHA 2, вопросы считывания/записи программы из/в контроллер. Уделено внимание вопросам разработки микропроцессорных систем управления, систем релейного, пропорционального и пропорционально-интегрально-дифференциального управления (ПИД-управления). Подробно рассмотрены структура двигателя постоянного тока и вопросы управления двигателями постоянного тока.

Пособие предназначено для закрепления и углубления теоретических знаний, приобретения практических навыков работы в области микропроцессорного управления и создания программ для микропроцессорной техники.

УДК 004.31-022.53(076)

ББК 32.973.26-04я73

ISBN 978-985-543-200-6

© УО «Белорусский государственный  
университет информатики  
и радиоэлектроники», 2016

## СОДЕРЖАНИЕ

### **Практическое занятие №1**

Основы работы с ПО AL-PCS/WIN-E и создание программ для контроллеров серии ALPHA и ALPHA 2..... 4

### **Практическое занятие №2**

Методы программного управления техническими системами..... 34

### **Практическое занятие №3**

Программное управление двигателем постоянного тока..... 55

Библиотека БГУИР

## **Практическое занятие №1**

### **Основы работы с ПО AL-PCS/WIN-E и создание программ для контроллеров серии ALPHA и ALPHA 2**

#### **Цель работы:**

- 1) ознакомиться с конструкцией, назначением, возможностями программируемых логических контроллеров (ПЛК) семейства ALPHA 2;
- 2) изучить возможности среды программирования и обмена данными с ПЛК ALPHA 2;
- 3) изучить процессы записи программы в память ПЛК и запуска выполнения;
- 4) получить навыки создания программ в среде программирования AL-PCS/WIN-EU.

#### **1.1 Основы работы с программным обеспечением AL-PCS/WIN-E**

Данное практическое занятие посвящено изучению принципов программирования промышленных контроллеров серии ALPHA 2 производства фирмы Mitsubishi.

Контроллер серии ALPHA 2 может использоваться для выполнения функций управления как в условиях производства, так и офиса или дома. Контроллер выполняет цикл включения/выключения в выходных цепях электрического оборудования в соответствии с разработанной программой. Программирование осуществляется методом визуального программирования, при котором используются линии, соединяющие функциональные блоки в окне программирования. Программное обеспечение AL-PCS/WIN-E для Windows способно создавать и сохранять программу работы микроконтроллера.

#### **Главные особенности программного обеспечения AL-PCS/WIN-E**

AL-PCS/WIN-E – эффективный инструмент для программирования простых прикладных контроллеров серии ALPHA и ALPHA 2 (далее они будут называться серией контроллеров ALPHA, или контроллерами) в стиле функциональных блоков. Визуальный характер программного обеспечения помогает пользователю видеть и понимать связи между всеми частями программы. Эффективное и простое в использовании программное обеспечение AL-PCS/WIN-E имеет следующие удобные для пользователя возможности:

- *Дружественное к пользователю программирование.* Программы разрабатываются несложно, с использованием визуального подхода. Программное обеспечение позволяет работать в простом в использовании многооконном формате Windows.

• **Мониторинг и принудительная установка состояний ON/OFF.** Мониторинг в режиме реального времени AL-PCS/WIN-E позволяет тестировать двоичные устройства.

• **Функция эмуляции.** Используя функцию эмуляции, можно проверять работу программы без подключения аппаратного обеспечения.

• **Функция дистанционного технического обслуживания,** позволяющая загружать и выгружать программу контроллера серии ALPHA для контроля по телефонной линии.

• **Мониторинг в окне схематического представления.** Клиент может контролировать работу своей системы в окне мониторинга в схематическом представлении.

• **Пользовательский функциональный блок.** Пользователь может создавать специализированные функциональные блоки, комбинируя исходные функциональные блоки.

### 1.1.1 Вид рабочего экрана

Для информации пользователя на рисунке 1.1 указаны панели инструментов, которые можно включить/отключить в строке меню при помощи вкладки Вид.

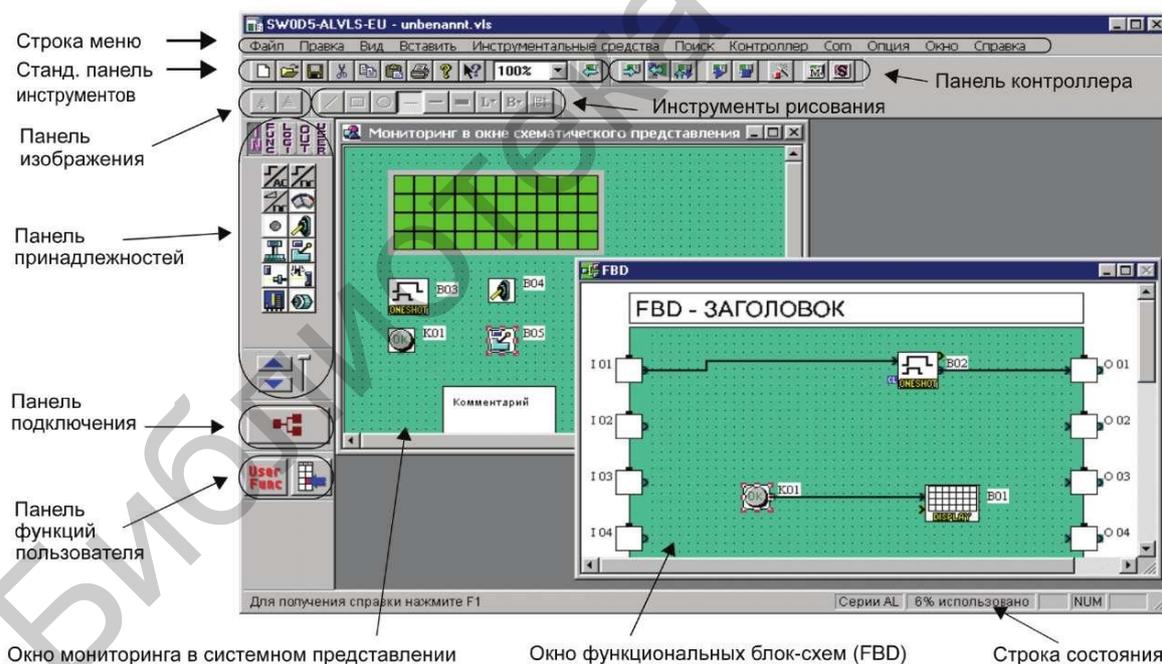


Рисунок 1.1 – Вид рабочего экрана

В **строке меню** находятся вкладки Файл, Правка, Вид, Вставить, Инструментальные средства, Поиск, Контроллер, Сот, Опция, Окно и Справка.

**Стандартная панель инструментов** содержит кнопки Новый, Открыть, Сохранить, Вырезать, Вставить, Печатать, О программе, Контекстная справка, Изменение размера окна и Прочитать информацию контроллера.

**Панель изображения** содержит кнопки Импортировать и Экспортировать. Кнопка Импортировать позволяет вставить предварительно экспортированный файл изображения (\*.img) в окно Мониторинг в окне схематического представления.

**Панель инструментов рисования** содержит кнопки Рисование линий, Рисование прямоугольников, Рисование овалов, Тонкая линия, Средняя линия, Толстая линия, Цвет линии, Цвет кисти и Выравнивание объектов.

**Инструментальная панель контроллера** содержит кнопки Запись в контроллер, Сверить данные контроллера, Диагностика контроллера, Запуск контроллера, Остановить контроллер, Мастер Авто FBD, Запуск/останов мониторинга и Запуск/останов моделирования.

**Инструментальная панель принадлежностей** содержит кнопки Входные сигналы, Функции, Логические функции, Выходные сигналы и Функции пользователя (таблица 1.1).

Таблица 1.1 – Инструментальная панель принадлежностей

<b>Кнопки панели принадлежностей</b>	<b>Описание</b>
Входные сигналы	Содержит значки для входных сигналов и системных битов
Функции	Содержит значки для функциональных блоков
Логические функции	Содержит значки для логических функциональных блоков
Выходные сигналы	Содержит значки для выходных сигналов и управляющих битов
Функции пользователя	Содержит значки для зарегистрированных функций пользователя

**Инструментальная панель подключения** представлена единственной кнопкой Подключение.

**Инструментальная панель функций пользователя** содержит две кнопки: User Func (Функция пользователя) и Регистрация функции пользователя.

**Строка состояния** отображает текущее состояние используемой серии, процентное использование функционального блока, клавиши Caps Lock, клавиши Num Lock и клавиши Scroll Lock.

### 1.1.2 Окно функциональной блок-схемы (FBD)

Окно функциональной блок-схемы (рисунок 1.2) используется для программирования контроллеров серии ALPHA. Окно FBD включает большой прямоугольник (по умолчанию зеленого цвета) – область подключений, которая

известна также как база размещения функциональной блок-схемы. В верхней части окна находится диалоговое окно заголовка; с правой и левой стороны от базы размещения вертикально расположены прямоугольники входов и выходов соответственно. При создании программы для контроллеров серии ALPHA программируемые компоненты помещаются на базу размещения или в прямоугольники и соединяются одинарными проводами. Окно FBD также известно как область подключений FBD.

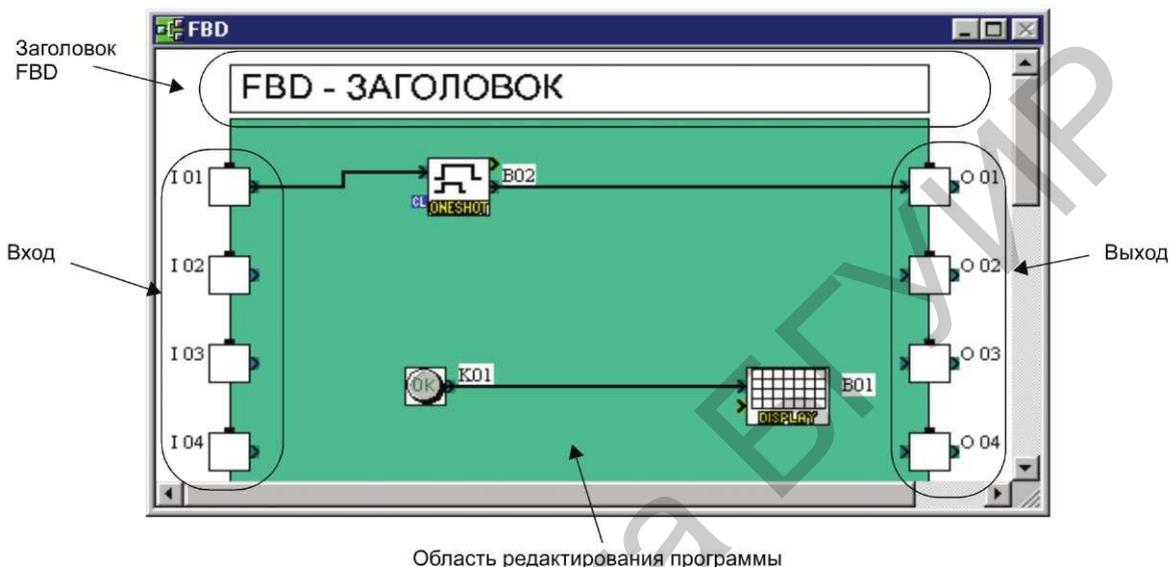


Рисунок 1.2 – Окно функциональной блок-схемы

Размер области редактирования может изменяться мышью. Используя экран FBD, пользователь имеет возможность выполнять следующие десять операций:

- 1) помещать сигналы ввода-вывода и функции, используя инструментальную панель принадлежностей;
- 2) задавать параметры для функций;
- 3) легко выполнять подключения различных компонентов с помощью анализатора подключений;
- 4) записывать алгоритм программы и информацию об устройствах ввода-вывода в контроллер серии ALPHA;
- 5) вызывать Мастер Авто FBD, облегчающий процесс создания схем начинающим пользователям;
- 6) проверять алгоритм программы с помощью внутренних устройств (сигналов входа и выхода, помещенных в области подключений функциональной блок-схемы);
- 7) имитировать и проверять алгоритм программирования в автономном режиме, без реальных аппаратных средств контроллера серии ALPHA:
  - принудительно устанавливать/сбрасывать входные сигналы (ON/OFF);

- изменять параметры функций (таймеров, счетчиков, аналоговых данных и т. д.);
  - показывать на экране комментарии или значения функций;
  - контролировать состояния компонентов, изменяя цвет соединительных проводов (например, красный провод – OFF, синий провод – ON);
- 8) считывать информацию из контроллера серии ALPHA и обновлять программу на экране FBD;
  - 9) наблюдать за контроллером серии ALPHA в режиме онлайн;
  - 10) получать распечатку экрана FBD и другой размещенной на экране информации.

### **1.1.3 Контроль в окне схематического представления системы**

Контроль в схематическом представлении основан на использовании модифицированного окна мониторинга. Это окно позволяет показать значки для ввода-вывода, функциональные блоки, изображения, а также состояние ЖК-дисплея контроллера.

Размер области редактирования в окне схематического представления может изменяться мышью. В окне схематического представления пользователь имеет возможность выполнять следующие операции:

- 1) создавать собственные схемы, используя Панель инструментов рисования;
- 2) размещать устройства ввода-вывода и функциональные блоки, создавая структуру программы для контроллера серии ALPHA;
- 3) помещать на экране схематического представления компоненты OLE-технологии (Object Linking and Embedding);
- 4) осуществлять мониторинг или тестировать контроллер серии ALPHA;
- 5) имитировать и проверять алгоритм программирования в отсутствие реальных аппаратных средств построения контроллера серии ALPHA;
- 6) получать распечатку окна мониторинга в схематическом представлении и другой размещенной на экране информации;
- 7) импортировать растровые изображения.

### **1.1.4 Режим программирования**

В этом режиме создаются все программы, а также добавляются и удаляются компоненты системы. Режим программирования также известен как режим составления схемы, когда все панели инструментов и опции меню используются для программирования или составления схем. При создании новых или открытии готовых файлов окно FBD и окно схематического представления будут по умолчанию установлены в режим программирования. Соответственно, в окне FBD будет показан заголовок «FBD (функциональная

блок-схема)», а в окне схематического представления – «Мониторинг в окне схематического представления».

### **Окно FBD в режиме программирования**

Как следует из названия, в режиме программирования можно использовать все функции, необходимые для создания программы. В режиме программирования пользователь может добавлять функциональные блоки или другие сигналы, задавать или изменять параметры, перемещать компоненты, а также выполнять подключения.

Функция Мастера Авто FBD имеется и в этом режиме, ее основная цель – ознакомить неопытных пользователей с сигналами, функциональными блоками, подключением и другими приемами программирования.

### **Мониторинг в окне схематического представления в режиме программирования**

В окне схематического представления пользователь может добавлять и перемещать объекты блок-схемы, а также изменять их размеры. Используя Инструментальную панель принадлежностей, пользователь сможет добавлять компоненты, имеющиеся на экране FBD. Возможно копирование компонентов на/с FBD экрана. Любой компонент в окне можно выбрать, щелкнув на нем левой кнопкой мыши, и переместить в желательное место. Двойной щелчок на функциональном блоке открывает диалог установки его параметров.

Пользователь может создавать объекты, используя Панель инструментов рисования с командами рисования линий, овалов и прямоугольников и/или импортируя объекты с помощью команд вкладки Вставить.

#### **1.1.5 Режим моделирования**

Режим моделирования позволяет имитировать условия, в которых будет выполняться программа, без физического подключения аппаратных средств. Это очень эффективный инструмент для отладки программы перед записью содержания программы в реальный контроллер серии ALPHA.

Информация, генерируемая программой, непрерывно считывается с имитатора. Программа AL-PCS/WIN-E производит постоянное обновление состояний и текущих значений сигналов и функциональных блоков; с другой стороны, пользователь может вводить значения самостоятельно с целью моделирования. Щелчком мыши пользователь может управлять состоянием (ON/OFF) входов и непосредственно устанавливать аналоговые значения.

#### **1.1.6 Режим мониторинга**

В этом режиме пользователь может вести наблюдение и тестировать аппаратные средства реального контроллера серии ALPHA, который подключен к ПК посредством интерфейса AL-232CAB или через модем.

Информация о состоянии и текущих значениях сигналов и функций непрерывно считывается с контроллера серии ALPHA и соответственно обновляется в AL-PCS/WIN-E.

Если состояние контроллера серии ALPHA обновлено с помощью управляющих кнопок контроллера, то появится окно сообщений, предупреждающее об обновлении состоянии контроллера ALPHA, и мониторинг будет остановлен. Программа автоматически перейдет в режим программирования.

## 1.2 Программирование функциональных блок-схем (FBD) контроллеров серии ALPHA 2

### 1.2.1 Открытие нового файла

Создавая новый файл, следуйте приведенным ниже инструкциям. Кроме того, можно открыть существующий файл, используя команду «Открыть...» из вкладки Файл или щелкнув на значке Открыть. Откроется список текущих файлов AL-PCS/WIN-E, которые по умолчанию хранятся в каталоге Prog. Выберите файл из этого каталога или воспользуйтесь просмотром, чтобы открыть файл в другом месте.

Чтобы открыть новый файл:

- 1) запустите AL-PCS/WIN-E;
- 2) нажмите кнопку Новый на Стандартной панели инструментов или выберите Файл, затем Новый в строке меню;
- 3) щелчком выберите название соответствующей серии контроллера;
- 4) щелчком выберите необходимый вариант входов/выходов;
- 5) нажмите ОК.

Значок цифрового входа, значок аналогового входа, значок входа для AS-интерфейса, системный бит, операционная клавиша

Значок цифрового входа, значок выхода для AS-интерфейса, управляющий бит

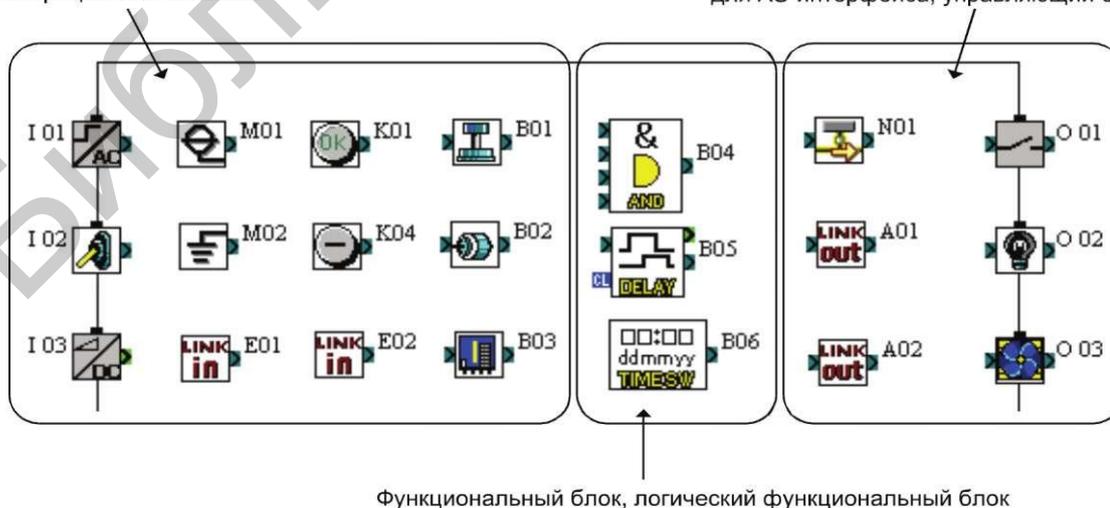


Рисунок 1.3 – Компоненты программы

## 1.2.2 Компоненты (значки) программы

Программы для контроллеров серии ALPHA могут создаваться и редактироваться в режиме редактирования программы, который выбирается в AL-PCS/WIN-E по умолчанию. Программирование контроллеров серии ALPHA производится путем размещения компонентов на базе функциональных блок-схем (FBD) и формирования соединений между различными компонентами с использованием анализатора подключений. Имеются 8 типов системных компонентов, которые можно поместить на базу функциональных блок-схем (FBD). Ниже приведено краткое описание функций и мест для размещения различных компонентов.

### База функциональных блок-схем (FBD)

База функциональных блок-схем (FBD) на экране FBD представляет собой платформу, на которой строится программа для контроллеров серии ALPHA.

### Входы

К входным сигналам относятся цифровые входы, аналоговые входы и биты AS-интерфейса. Значки находятся на панели входных сигналов инструментальной Панели принадлежностей IN слева от базы функциональных блок-схем (FBD).

Цифровые сигналы можно поместить на прямоугольники входов слева от базы функциональных блок-схем (FBD) или на саму базу. Принцип работы каждого значка одинаков. Все они были разработаны для помощи в документировании и понимании программы. Используйте значок входа, соответствующий реальному устройству ввода, которое будет подключено к контроллеру серии ALPHA. Если поместить курсор мыши на значок, будут показаны имена сигналов.

Аналоговые и температурные сигналы (от адаптера AL2-2PT-ADP или AL2-2TC-ADP) можно помещать только в прямоугольники входов 101–108.

### Выходы

К выходным сигналам относятся цифровые выходы и биты AS-интерфейса. Значки находятся на панели выходных сигналов инструментальной Панели принадлежностей OUT справа от базы функциональных блок-схем.

Выходы системы можно поместить на прямоугольники выходов справа от базы функциональных блок-схем (FBD). Выходы, помещенные вне прямоугольников, будут действовать как нейтральные блоки. Выходные блоки имеют один входной вывод и один выходной вывод, которые можно соединять с другими функциональными блоками или выходами.

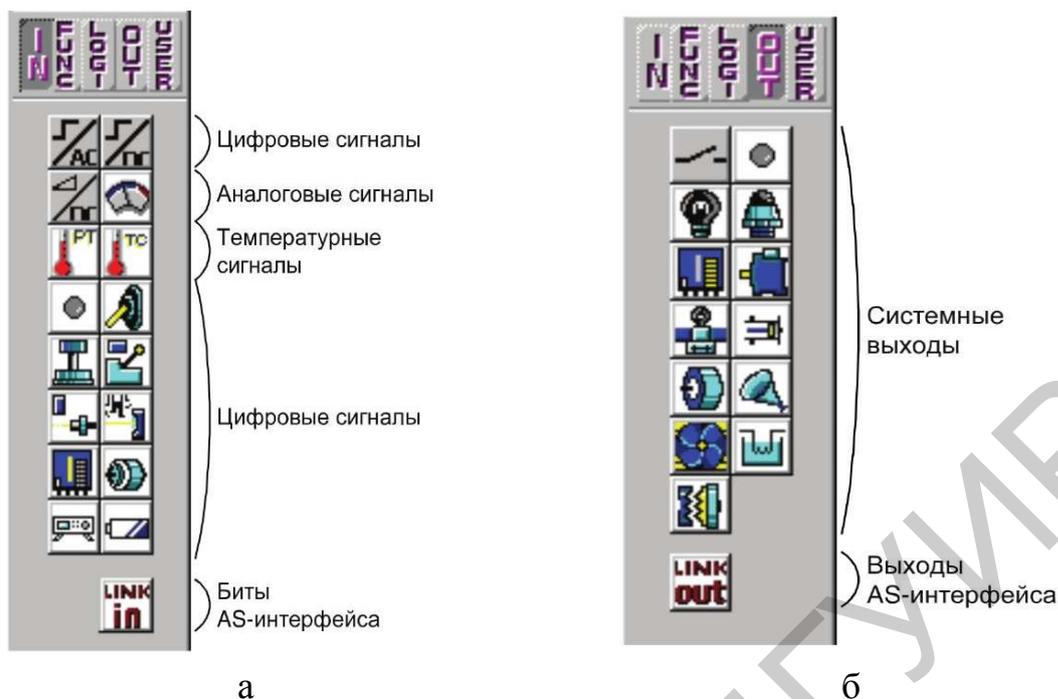


Рисунок 1.4 – Панель входных (а) и выходных (б) сигналов

Имеются 4 выхода AS-интерфейса, значок LINK OUT в A01–A04. Эти выходы можно поместить в любом месте на базе функциональных блок-схем (FBD); они не учитываются при подсчете количества выходов системы.

### Операционные клавиши

Имеются 8 сигналов, которым соответствуют клавиши на передней панели, и эти сигналы могут использоваться как дополнительные цифровые входы (рисунок 1.5). Это клавиши ESC, OK, +, –, T, < и ►. Чтобы активизировать сигналы, нажмите клавиши на передней панели, когда контроллер серии ALPHA находится в режиме РАБОТА. Эти клавиши могут быть помещены только на базу функциональных блок-схем (FBD); они находятся в панели входных сигналов инструментальной панели принадлежностей IN. Имена сигналов появятся, когда курсор мыши будет помещен на значок.

### Логические функциональные блоки

Логические функциональные блоки – И, ИЛИ, НЕ, Исключающее ИЛИ, И-НЕ, ИЛИ-НЕ – обладают только цифровыми входами. Выходы этих блоков переключаются в зависимости от состояния сигналов (ON/OFF) на входах. Логические блоки находятся в панели логических функций инструментальной Панели принадлежностей OUT. Имена блоков появятся, когда курсор мыши будет помещен на значок. Описание блоков представлено в таблице 1.2.

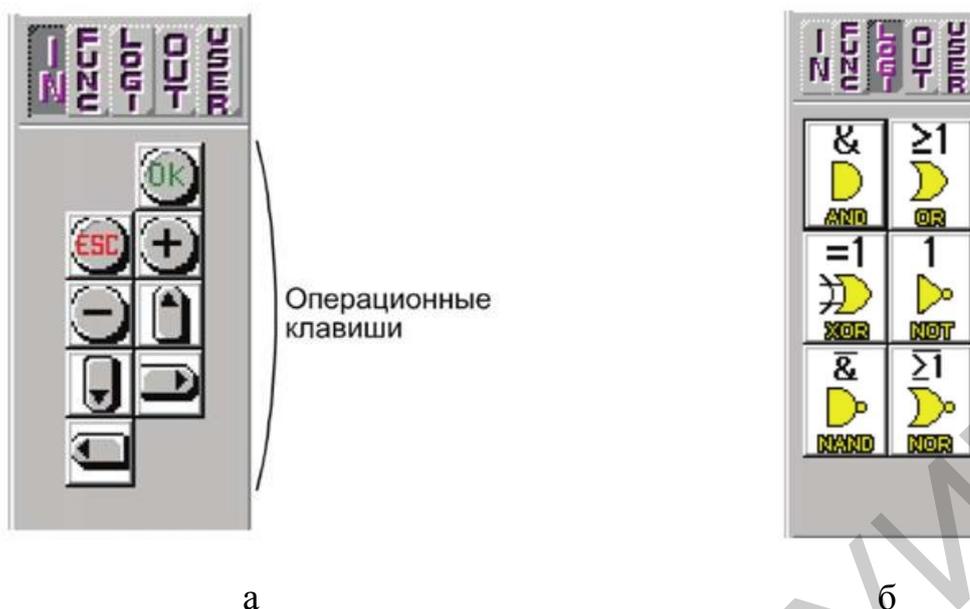


Рисунок 1.5 – Панели Операционные клавиши (а) и Логические функциональные блоки (б)

Таблица 1.2 – Логические функциональные блоки

Логический функциональный блок	Описание
И	Состояние выхода ON, если состояние всех входов ON. Неиспользуемые входы рассматриваются как ON
ИЛИ	Состояние выхода ON, если состояние хотя бы одного входа ON. Неиспользуемые входы рассматриваются как OFF
НЕ	Инвертирует сигнал; изменяет OFF на ON и наоборот
Исключающее ИЛИ	Состояние выхода будет ON, когда только один из двух входов находится в состоянии ON
И-НЕ	Состояние выхода OFF, если состояние всех входов ON. Неиспользуемые входы рассматриваются как ON
ИЛИ-НЕ	Состояние выхода OFF, если состояние хотя бы одного входа ON. Неиспользуемые входы рассматриваются как OFF

### Функциональные блоки

Вид поддерживаемых функциональных блоков зависит от серии контроллера, как показано в таблице 1.3.

Функциональные блоки находятся на панели функций инструментальной Панели принадлежностей FUNC. Имена блоков появятся, когда курсор мыши будет помещен на значок.

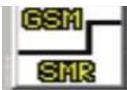
Таблица 1.3 – Функциональные блоки

Функциональный блок		Описание	Изображение кнопки
Название кнопки	Символ		
1	2	3	4
Булева функция (Boolean)	BL	Логическое уравнение, использующее И, ИЛИ, Исключающее ИЛИ, НЕТ	
Установка/Сброс (Set/Reset)	SR	Фиксирует реле в состоянии установка или сброс, задает приоритет Установки или Сброса	
Задержка (Delay)	DL	Обеспечивает задержку сигнала по включению, выключению или обоим фронтам	
Однократное исполнение (One Shot)	OS	Формирует одиночный импульс, используя заданную продолжительность или входной сигнал. Имеется вывод входа сигнала сброса	
Импульс (Pulse)	PL	Формирует импульс по включению, выключению или обоим фронтам	
Замена, альтернативная функция (Alternate)	AL	Выходной сигнал переключается на каждый входной импульс	
Счетчик (Counter)	CN	Подсчитывает передние фронты входного сигнала, может сбрасывать по сигналу сброса	
Реверсивный счетчик (Up/Down Counter)	UD	Выполняет прямой или обратный счет передних фронтов входного сигнала. Когда подсчитанная величина достигает установленного значения, выходной сигнал переходит в состояние ON	
Мерцание (Flicker)	FL	Формирует серию импульсов согласно времени включенного/выключенного состояния, числу циклов, длительности	

Продолжение таблицы 1.3

1	2	3	4
Сравнение (Compare)	CP	Сравнивает два значения с точки зрения $<$ , $>$ , $=$ , $\leq$ , $\geq$ , $\diamond$ (аналоговые значения, прямые установки или значения функционального блока)	
Реле времени (Time Switch)	TS	Использует часы реального времени для включения/выключения выходного сигнала на календарной или недельной основе	
Коэффициент усиления (Gain)	OG	Управляет аналоговыми значениями $y = A/Bx + C$ . Устанавливает значения верхнего и нижнего ограничителей	
Триггер Шмидта (Schmitt trigger)	ST	Выход переходит в состояние ON, когда входной сигнал превышает верхнее пороговое значение, и выключается (OFF), когда входной сигнал падает ниже нижнего порогового значения (или наоборот)	
Сравнение зон (Zone Compare)	ZC	Проверяет, находится ли входная величина в пределах зоны, ограниченной верхней и нижней границами (аналоговые значения, прямые установки или значения Функционального блока)	
Отображение (Display)	DP	Функция отображения пользовательского экрана (сообщения или данные) на LCD дисплее контроллера	
Счетчик моточасов (Hour Meter)	HM	Регистрирует суммарное время, когда сигнал имеет значение ON	
Определение скорости (Speed Detect)	SPD	Функция измеряет частоту входных импульсов (ON/OFF) на установленном периоде времени. Частота сравнивается с диапазоном значений, и выход включается/выключается согласно результату	
Модуляция ширины импульса	PWM	При запуске этого функционального блока формируется непрерывная серия импульсов. Задаются следующие параметры импульса: режим работы (непосредственная установка, аналоговый, значения Функционального блока), период	

Продолжение таблицы 1.3

1	2	3	4
Сложение	ADD	$y = A + B$	
Вычитание (Subtraction)	SUB	$y = A - B$	
Умножение (Multiplication)	MUL	$y = A \cdot B$	
Деление (Division)	DIV	$A : B = q, r$ (частное и остаток)	
Вычисление (Calculation)	CAL	Уравнение, использующее операторы +, -, *, /, % и выбранные данные	
Сдвиг (Shift)	SFT	Выходной сигнал будет установлен в соответствии с состоянием входного сигнала непосредственно перед тем, как входной сигнал сдвига будет установлен в ON. При использовании 8-разрядной операции сдвига функциональный блок Сдвиг должен использоваться 8 раз	
Задержанная замена (Delayed Alternate)	DAL	После времени задержки выходной сигнал переключается на каждый входной импульс	
Аналоговый выход (Analog Output)	AO	Выводит аналоговое значение в указанном канале на установленной плате расширения AL2-2DA	
ПИД (PID)	PID	ПИД-регулятор	
Отобразить (Display)	CDP	Управляет отображаемой информацией на LCD-дисплее контроллера	
Connect	_B	Этот функциональный блок CONNECT является внутренним функциональным блоком, показывающим память, которая используется системными битами, битами для AS-интерфейса, и операционными клавишами	

## 1.3 Создание функциональных блок-схем (FBD)

### 1.3.1 Размещение значков функциональных блок-схем

Значки функциональных блок-схем представлены на рисунке 1.6. Значок можно выбрать, щелкнув левой клавишей мыши на инструментальной Панели принадлежностей, и разместить, щелкнув в нужном месте на базе функциональных блок-схем. Кроме того, Функциональные блоки и Логические функциональные блоки можно разместить, используя команду «Вставить функцию» из вкладки Правка.

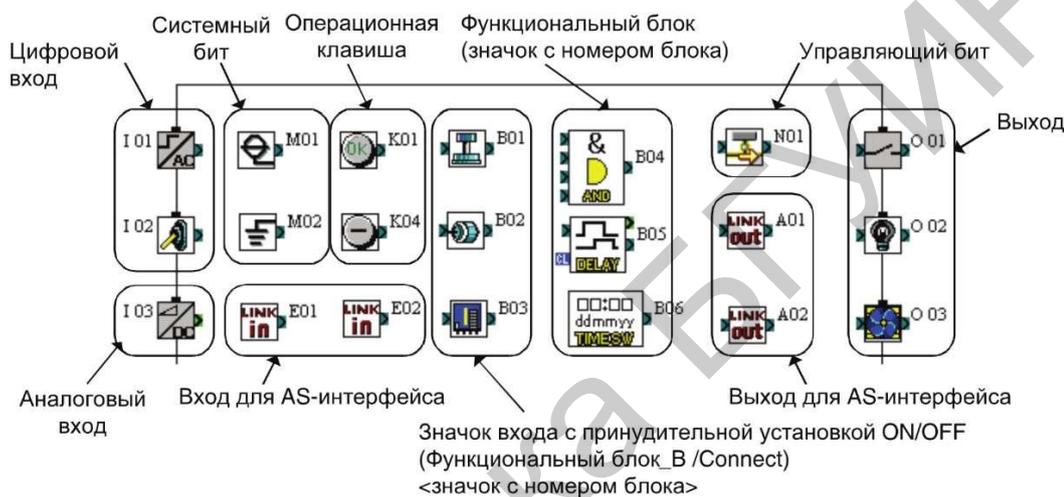


Рисунок 1.6 – Значки функциональных блок-схем

### Выбор типа контроллера

Количество входов/выходов и серию контроллера на базе функциональных блок-схем можно изменять в окне «Выбор типа контроллера» (рисунок 1.7), которое выбирают, используя вкладку Опция.

Кроме того, это окно появляется, когда открывается новый файл.

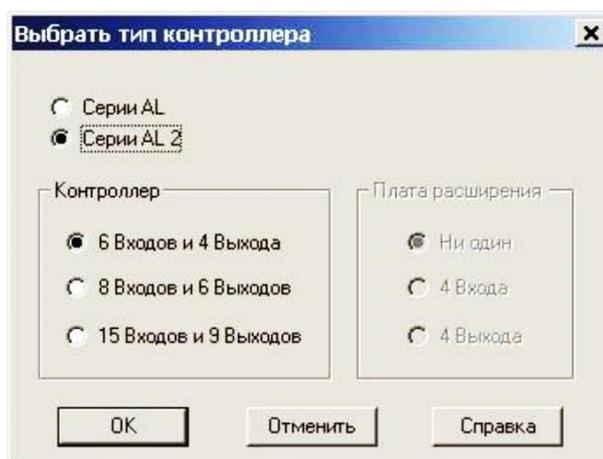


Рисунок 1.7 – Окно выбора типа контроллера

### 1.3.2 Соединения (провода) между значками

#### Входные и выходные выводы

Небольшие прямоугольники на левой и правой стороне значка называются выводами. Имеются 4 типа этих выводов: вывод цифрового входа (сброса), вывод цифрового выхода, входной вывод типа слова и выходной вывод типа слова. Выводы, расположенные на левой стороне значка, относятся к выводам ВХОДОВ, расположенные на правой стороне значка, – к выводам ВЫХОДОВ (рисунок 1.8).

Вывод типа слова маркируется светло-зеленым цветом на экране AL-PCS/WIN-E, чтобы отличить его от битового вывода.



Рисунок 1.8 – Входные и выходные выводы

#### Соединения (провода) между значками

Инструмент подключения графически (в виде линий с завершениями) отображает соединения, сделанные между входами, выходами, сигналами и функциями, представленными на экране FBD.

Инструмент подключения можно вызвать, щелкнув на инструментальной панели подключения. Она находится слева от окна FBD.

Чтобы сделать подключение между значками:

- 1) щелкните на инструментальной панели подключения;
- 2) щелкните на входном или выходном выводе на значке, чтобы подключить провод;
- 3) щелкните на выходном или входном выводе целевого значка, чтобы сделать подключение между двумя значками;
- 4) если выводы не будут физически соединены, возвратитесь к пункту 2;
- 5) панель подключения останется активизированной до тех пор, пока пользователь не щелкнет левой кнопкой мыши по любой другой инструментальной панели или на свободной области в окне FBD.

На рисунке 1.9 указана схема соединения блоков.

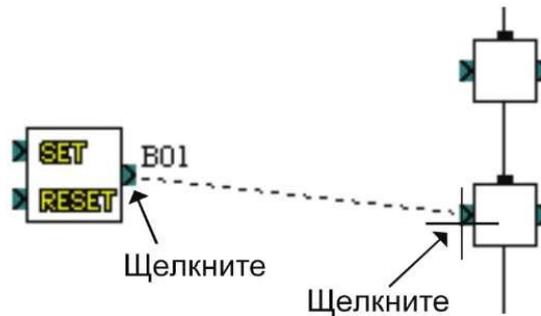


Рисунок 1.9 – Соединение блоков

### Настройки значков

Диалоговое окно всех значков включает поле Комментарий, флажок «Отобразить комментарий», флажок «Отобразить информацию мониторинга» и флажок «Отобразить номер сигнала». Это диалоговое окно можно открыть двойным щелчком на значке. Введенный комментарий будет показан на экране, только когда установлен флажок «Отобразить комментарий».

Точно так же номер сигнала будет показан на экране, только если установлен флажок «Отобразить номер сигнала».

### Параметры функциональных блоков

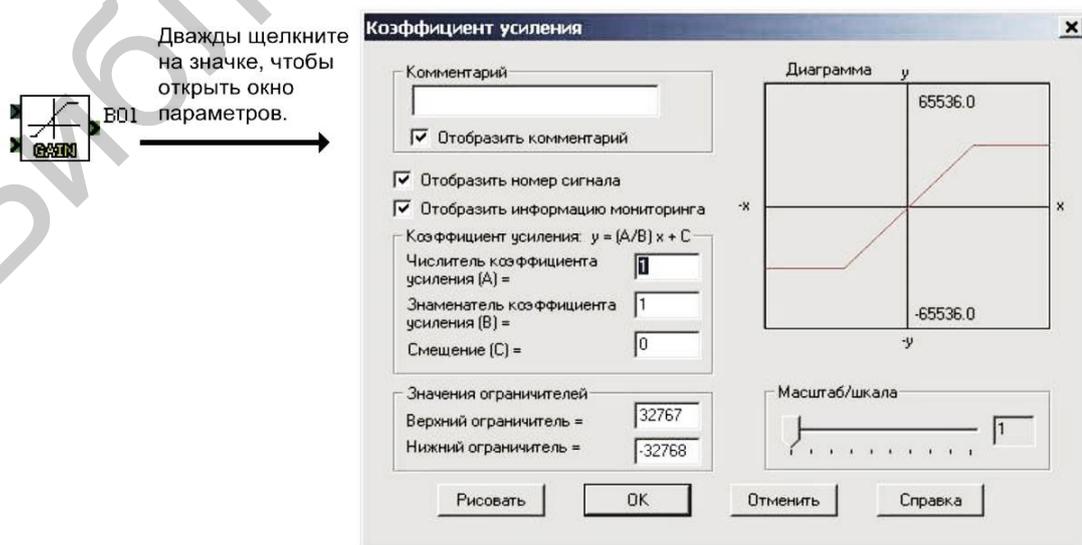


Рисунок 1.10 – Задание параметров функциональных блоков

## 1.4 Регистрация пользовательского функционального блока

Параметры функционального блока можно просмотреть, дважды щелкнув на функциональном блоке (рисунок 1.10). Откроется всплывающее диалоговое окно, в котором будут показаны редактируемые параметры и поле комментария для упрощения документации. Каждый функциональный блок имеет собственный набор редактируемых параметров, включая различные флажки отображения, показанные справа.

Пользовательский функциональный блок – это пользовательская комбинация стандартных функций, сохраненная для повторного использования.

Чтобы зарегистрировать пользовательский функциональный блок:

1) нажмите кнопку User Func (Функция пользователя) (рисунок 1.11) на панели функций пользователя и щелкните на базе функциональных блок-схем;

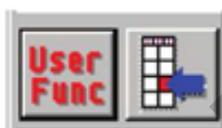


Рисунок 1.11 – Изображение кнопки User Func

2) установите параметры в окне вложенной функциональной блок-схемы (рисунок 1.12);

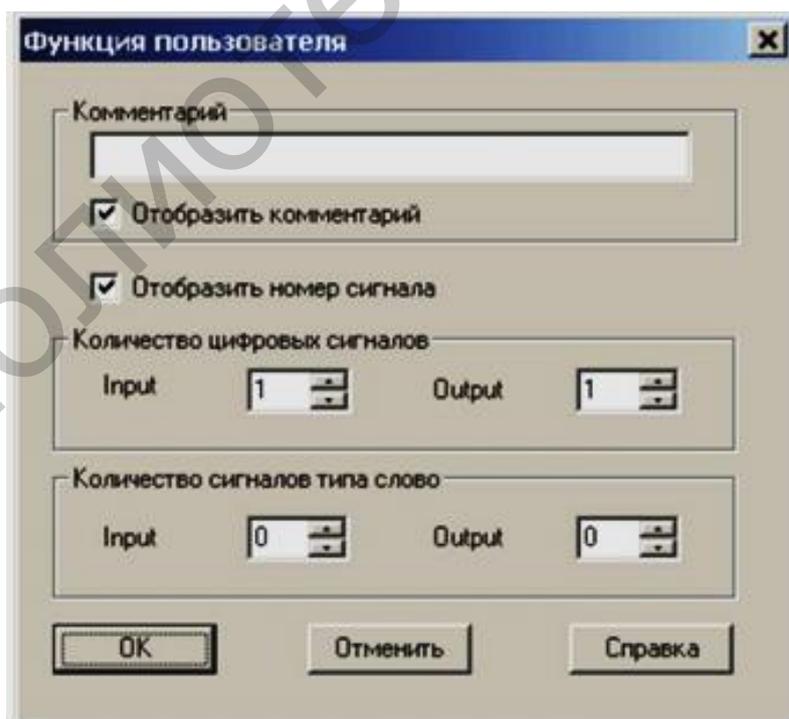


Рисунок 1.12 – Окно вложенной функциональной блок-схемы

3) создайте значок для пользовательского функционального блока (рисунок 1.13);

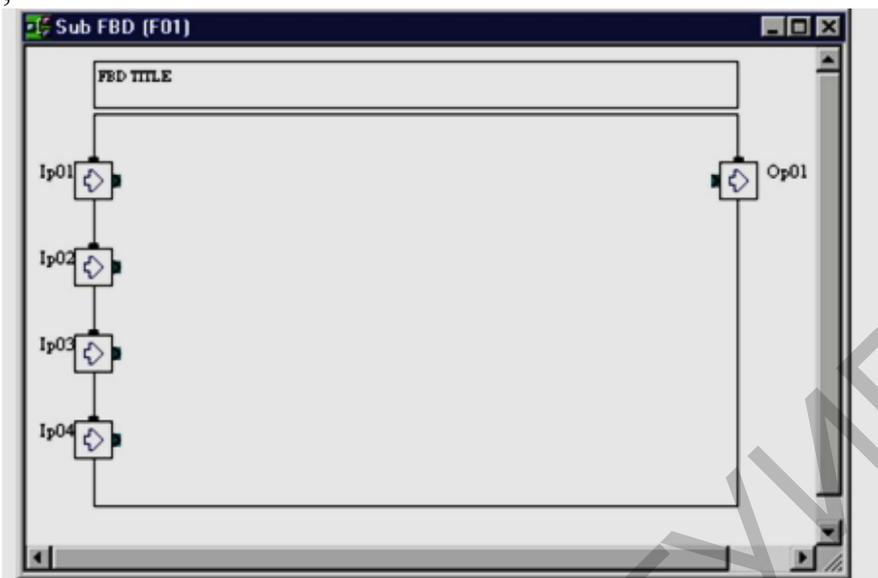


Рисунок 1.13 – Создание пользовательского функционального блока

- 4) подсоедините каждый значок;
- 5) установите параметры для функциональных блоков;
- 6) нажмите кнопку Регистрация функции пользователя, чтобы зарегистрировать пользовательский функциональный блок;
- 7) определите значки для состояний OFF и ON нового функционального блока согласно их спецификации (таблица 1.4):
  - нажмите кнопку Browse..., чтобы зарегистрировать каждый значок;
  - выберите изображение для ON или OFF из окна Open, затем нажмите кнопку Open, чтобы зарегистрировать значки ON или OFF.

Таблица 1.4 – Спецификация значков ON/OFF

Параметр	Технические данные
Тип файла	Файл точечной графики
Размер	36×36 пикселей
Цвет	16 цветов и 256 цветов

При регистрации растрового изображения (\*.bmp) для значков ON или OFF AL-PCS/WIN-E генерирует изображения \*\_ON.bmp или \*\_OFF.bmp для зарегистрированного функционального блока пользователя («\*» является именем зарегистрированного функционального блока пользователя);

8) нажмите кнопку ОК, чтобы завершить регистрацию пользовательского функционального блока.

При регистрации пользовательского функционального блока файлы \*.ruf, \*\_ON.bmp и \*\_OFF.bmp должны сохраняться в следующем каталоге (таблица 1.5).

Таблица 1.5 – Генерация файлов для пользовательского функционального блока

Имя файла	Каталог хранения	Описание
*.ruf	Установленный приложением каталог \Library\UserFunc например, C:\Program Files\Alvls\Library\UserFunc)	Информационный файл для пользовательского функционального блока
*_ON.bmp	Установленный приложением каталог \Library\ Bitmap например, C:\Program Files\Alvls\Library\Bit- map)	Изображение значка ON для пользовательского функционального блока
*_OFF.bmp	Установленный приложением каталог \Library\ Bitmap например, C:\Program Files\Alvls\Library\Bit- map)	Изображение значка OFF для пользовательского функционального блока

### Экспорт пользовательского функционального блока

Данные зарегистрированной функции пользователя (функционального блока пользователя) экспортируются (сохраняются) в указанный каталог. Это дает возможность регистрировать функцию пользователя (функциональный блок пользователя) с одного ПК на другой. Для экспорта выполните следующие действия:

- 1) щелкните на значке экспортируемой функции пользователя;
- 2) выберите команду «Экспортировать зарегистрированную функцию пользователя» из вкладки Файл;
- 3) в диалоговом окне «Экспортировать зарегистрированную функцию пользователя» выберите каталог, в котором будет храниться экспортированный файл и введите имя функции пользователя, затем нажмите кнопку Save.

Файлы ruf, \*\_ON.bmp и \*\_OFF.bmp должны храниться вместе в выбранном каталоге.

### Импорт пользовательского функционального блока

Данные зарегистрированной функции пользователя (функционального блока пользователя) импортируются из указанного каталога. Это дает возможность регистрировать/переносить функцию пользователя (функциональный блок пользователя) с одного ПК на другой. Для импорта выполните следующие действия:

- 1) выберите команду «Импортировать зарегистрированную функцию пользователя» из вкладки Файл;

2) в диалоговом окне «Импортировать зарегистрированную функцию пользователя» задайте любой каталог, в котором будут храниться импортируемые данные, и введите (выберите) имя импортированных данных, затем нажмите кнопку Open.

*Замечание.* Импортированные данные (\*.ruf, \*\_ON.bmp и \*\_OFF.bmp файлы) должны храниться в одном каталоге.

## 1.5 Окно мониторинга в схематическом представлении

Окно мониторинга в схематическом представлении может настраиваться пользователем, как показано на рисунке 1.14.

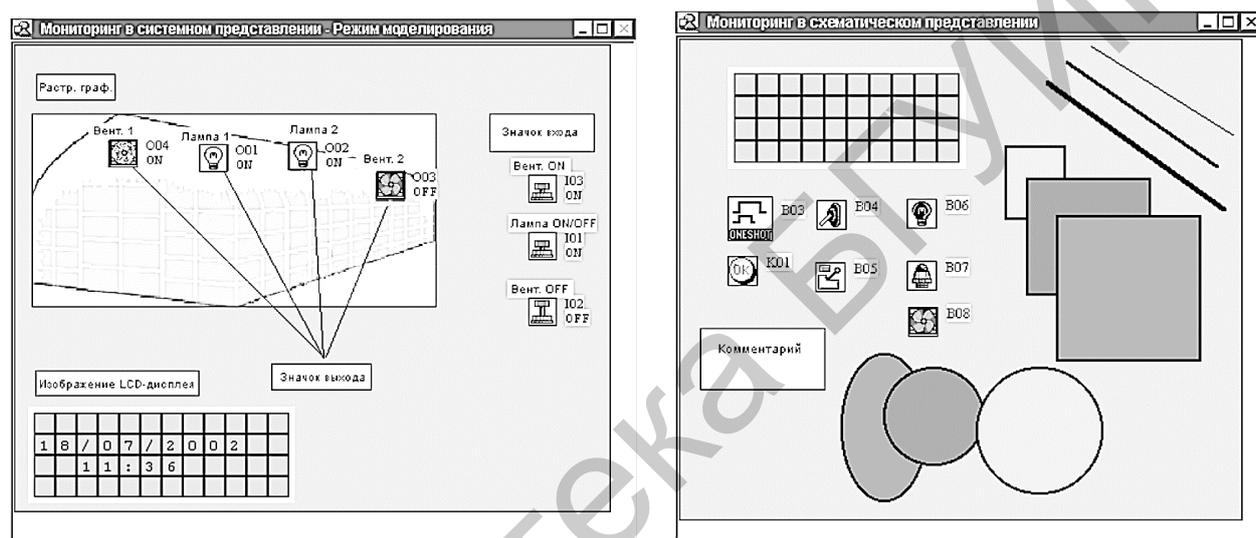


Рисунок 1.14 – Окно мониторинга

*Замечание.* Когда это окно активно, могут использоваться следующие инструментальные панели и функции:

1. Активные функции на Панели инструментов:

- инструментальные панели изображения и инструментов рисования;
- «Изображение LCD» и «Вставить новый объект...» вкладки Инструментальные средства.

2. Неактивные функции на Панели инструментов:

- инструментальные панели подключения и функций пользователя;
- «Мастер Авто FBD» вкладки Инструментальные средства;
- «Выбор типа контроллера...» вкладки Опция.

### 1.5.1 Изменение размеров базы в окне схематического представления

Когда окно мониторинга в схематическом представлении активно, база имеет минимальный размер. Можно изменить как ширину, так и высоту базы, перетаскивая правый или нижний края базы соответственно. Перетаскивая

мышью правый нижний угол прямоугольника базы, можно регулировать как высоту, так и ширину. Нельзя изменить размеры базы, перемещая левый и верхний края.

### **1.5.2 Изменение цвета базы функциональных блок-схем и значков**

#### **Изменение цвета базы**

Цвет базы функциональных блок-схем можно изменять с помощью команды «Установить цвет базового прямоугольника» вкладки Опция (цвет по умолчанию: зеленый с желтым оттенком).

#### **Настройка значков**

Можно использовать значки сигналов, зарегистрированных пользователем, и изображение для значка функционального блока. Значок можно зарегистрировать командой «Значки, определяемые пользователем» вкладки Опция. Пользователь может выбрать зарегистрированный значок командой «Набор значков функций» вкладки Опция.

### **1.5.3 Рисование линий, овалов и прямоугольников**

Линии можно рисовать, щелкнув на значке линии, и затем соответственно перемещая мышью. Щелкните кнопкой мыши, чтобы начать линию, и отпустите кнопку, достигнув желательной конечной точки.

Овалы и прямоугольники можно рисовать аналогичным образом, выбрав соответствующие значки.

#### **Перемещение и измерение размеров линий, овалов и прямоугольников**

Объекты можно перемещать в пределах окна схематического представления системы, щелкнув на них и перетаскивая в новую позицию. Чтобы изменить размеры, нажмите левую кнопку мыши на любом из краев и перемещайте его мышью. Курсор мыши изменит форму, превратившись в двойную стрелку, переходя через край объекта, указывая направление, в котором можно изменить его размер.

Размеры объектов можно изменять только в пределах окна схематического представления системы.

Функции Вырезать, Копировать и Вставить могут выполняться над объектами в пределах окна схематического представления системы.

#### **Изменение цветов**

Чтобы изменить цвет существующей линии, выберите линию левой кнопкой мыши. Щелкните на значке Цвет линии и выберите новый цвет для линии. Линия будет нарисована этим цветом. Этот цвет появится как цвет линии контура для овалов и прямоугольников. Цвет заливки для

прямоугольников и овалов можно изменить, щелкнув на значке Цвет кисти и выбрав новый цвет.

### **Изменение ширины линии**

Ширину существующей линии можно изменить, выбрав линию в окне схематического представления системы и щелкнув на значке желательной ширины линии – тонкой, средней или толстой. Ширина линии для овалов и прямоугольников изменяется аналогичным образом. Выбранная ширина станет активной шириной линии для всех инструментов рисования.

## **1.5.4 Изображение LCD-дисплея**

LCD-дисплей можно вставить в окно схематического представления системы либо с использованием вкладки Вставить, либо в ходе Моделирования или Мониторинга.

### **Вставка изображения LCD-дисплея**

Для активизации команды «Вставить изображение LCD» в базе функциональных блок-схем должен существовать функциональный блок отображения Display. Щелкните на команде, чтобы вставить LCD-дисплей в окно схематического представления системы. Если он уже присутствует в окне схематического представления, появится диалоговое окно ошибки.

Функциональный блок отображения Display вызовет диалоговое окно при переходе в режим Моделирование или Мониторинг. В этом окне будет выдан запрос, добавить ли дисплей на базу окна схематического представления системы. Выберите «Да», чтобы автоматически добавить LCD-дисплей.

## **1.6 Режим моделирования**

Режим моделирования позволяет пользователю запускать систему программирования автономно, без подключения фактических аппаратных средств контроллера. В ходе моделирования можно вносить изменения в параметры функционального блока, при необходимости вводить аналоговые значения и вручную переключать входные сигналы в состояния ON/OFF.

### **1.6.1 Отображение значков сигналов, проводов и функциональных блоков**

#### **Значки входов и выходов**

Все значки входов и выходов будут маркированы их номерами (101, 002, МОЗ, К04, Е01, А01 и т. д.) и состоянием ON/OFF.

## **Провод**

Провода, подключенные к цифровым выводам, для выходных выходов в состоянии ON становятся красными, а для выходных выводов в состоянии OFF – синими. (Эти цвета можно изменять во вкладке «Опции» в режиме программирования).

## **Функциональный блок**

Все функциональные блоки будут маркированы их номерами (B01, B02, B03 и т. д.), состоянием ON/OFF, значениями таймера, значениями счетчика, настройками приоритета и значениями прочих параметров. Вся важная информация о каждом компоненте видна с первого взгляда.

### **1.6.2 Запуск режима моделирования**

Когда программа готова к проверке, перейдите в режим моделирования, щелкнув на значке Запуск/останов моделирования на инструментальной Панели контроллера или используя опцию «Моделирование/Запуск» вкладки Контроллер. При необходимости вы можете увидеть окно FBD или окно мониторинга в схематическом представлении.

## **1.7 Считывание/запись программы из/в контроллер**

### **1.7.1 Запись программы в контроллер**

1. Щелкните на команде «Запись в контроллер» во вкладке Контроллер.
2. Не выбирайте опцию «Создать файл перемещаемой информации» без необходимости.
3. Выберите опцию «Запустить и начать мониторинг», если контроллер подключен.
4. Нажмите кнопку ОК, чтобы начать запись программы в контроллер серии ALPHA.

*Замечание.* Запись программы в контроллер серии ALPHA возможна только в том случае, когда контроллер находится в состоянии Останов.

### **1.7.2 Считывание программы из контроллера**

1. Щелкните на пункте «Считывание из контроллера» во вкладке Контроллер.
2. Щелкните на любой из кнопок переключателя:
  - отобразить данные автоматически: размещает значки соответствующим образом, не используя файл перемещаемой информации;
  - отобразить данные из пользовательского файла перемещаемой информации: для размещения значков используется файл перемещаемой информации указанной программы.

Если файл перемещаемой информации для этой программы найден в каталоге Relocation, это окно не открывается, и начинается считывание программы.

3. Нажмите кнопку ОК, чтобы начать считывание программы из контроллера.

*Замечание.* Файл перемещаемой информации имеет такой же формат, как любой другой .vis-файл, и содержит информацию о FBD и окне схематического представления, включая позиционную информацию. Данные, которые записаны в контроллер, содержат только информацию о FBD, исключая информацию о позиции. Этот файл создается после того, как данные были успешно записаны в контроллер, в каталоге Relocation. Таким образом, его можно использовать при выполнении команды «Считывание из контроллера».

## 1.8 Мониторинг

Режим мониторинга можно использовать, чтобы контролировать текущее состояние контроллера серии ALPHA, выполняющего программу. Контроллер серии ALPHA должен поддерживать связь с AL-PCS/WIN-E через AL-232CAB или модем. В противном случае при запуске режима мониторинга появится диалоговое окно ошибки.

Состояние контроллера серии ALPHA (Работа или Останов) будет постоянно контролироваться и отображаться, как показано на рисунке 1.15.

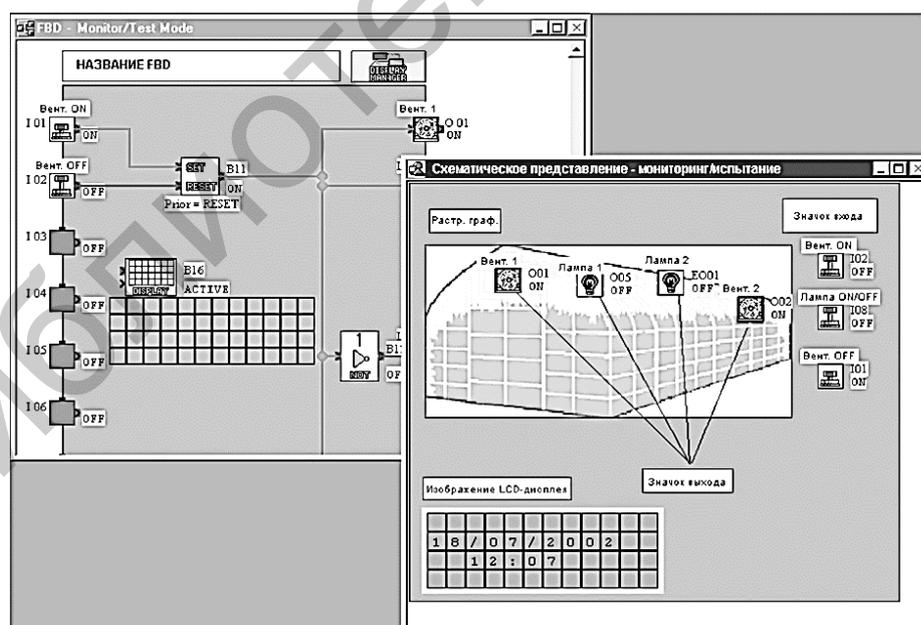


Рисунок 1.15 – Мониторинг состояния контроллера серии ALPHA

Перед началом мониторинга содержимое памяти AL-PCS/WIN-E и контроллера серии ALPHA должно быть одинаковым. Программа AL-PCS/WIN-E будет постоянно считывать содержимое контроллера серии ALPHA

и соответственно обновлять информацию. Изменения, сделанные в параметрах программы, будут передаваться в контроллер серии ALPHA и воздействовать на работу оборудования.

*Замечание.* Неактивные функции в режиме мониторинга. В режиме мониторинга будут деактивированы вкладки Правка, Вставить и Инструментальные средства, а также панели принадлежности, подключения и инструментов рисования. Находясь в режиме мониторинга, вы не сможете редактировать программу – добавлять, удалять, подключать или перемещать любые компоненты системы.

### 1.8.1 Запуск режима мониторинга

Чтобы войти в режим мониторинга, щелкните на значке Запуск/останов мониторинга или используйте команду «Мониторинг/Испытание/Запуск» вкладки Контроллер.

### 1.8.2 Установка/сброс сигналов (принудительная установка/сброс)

Выводы контроллера переводятся в состояние ON/OFF однократным щелчком на значке вывода (таблица 1.6).

Таблица 1.6 – Установка/сброс сигналов

Компоненты	Режим Работа	Режим Останов
Входы (I01 ~ I15, EI0 ~ EI4)	Замечание 1	Замечание 1
Выходы (O01 ~ O09, EO1 ~ EO4)	x Замечание 2	✓
Входы для AS-интерфейса (E01 ~ E04)	Замечание 1	Замечание 1
Выходы для AS-интерфейса (A01 ~ A04)	x Замечание 2	✓
Операционные клавиши (K01 ~ K08)	Замечание 1	1
Управляющие биты (N01 ~ N04)	x Замечание 2	✓
Выходной сигнал функционального блока	Замечание 3	x Замечание 4

*Замечание 1.* Входные выводы, входные биты AS-интерфейса и операционные клавиши можно принудительно устанавливать/сбрасывать, но состояние значка зависит от состояния (ON/OFF) аппаратных средств, поэтому изменение состояния (ON/OFF) сигнала нецелесообразно.

*Замечание 2.* Выходные выводы и выходные биты AS-интерфейса не могут принудительно устанавливаться/сбрасываться, потому что выход определяется состоянием программы.

*Замечание 3.* Все функциональные блоки можно принудительно устанавливать/сбрасывать в режиме RUN, но выход функционального блока может зависеть от состояния его входов, поэтому изменение состояния сигнала нецелесообразно.

*Замечание 4.* Не все функциональные блоки можно принудительно включать/сбрасывать.

### **1.8.3 Изменение параметров функционального блока**

Параметры функционального блока можно изменить, дважды щелкнув на функциональном блоке. Измените соответствующие параметры и нажмите кнопку ОК, чтобы принять изменения. На экране будут показаны новые значения и выполнены любые изменения согласно настройкам программы. Реле времени не могут редактироваться в режиме мониторинга.

### **1.8.4 Остановка режима мониторинга**

Пользователь может остановить режим мониторинга и перейти в режим программирования, нажав кнопку Запуск/останов мониторинга на инструментальной Панели контроллера или выбрав команду «Мониторинг/Испытание/Останов» во вкладке Контроллер.

## **1.9 Порядок выполнения**

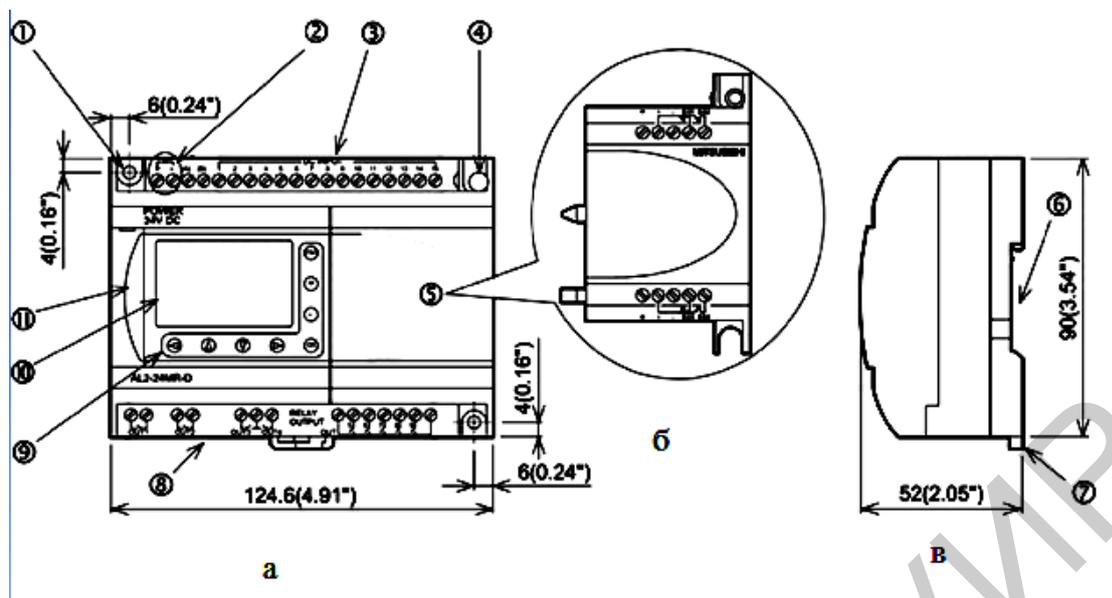
Перед работой с контроллером рекомендуется изучить следующие руководства:

1. ALPHA 2. Простой прикладной контроллер. Руководство по аппаратной части.

2. ALPHA 2. Простой прикладной контроллер. Руководство по программному обеспечению.

Для программирования контроллера необходим компьютер с COM-портом или переходник USB-RS232 (виртуальный COM-порт), кабель Mitsubishi для программирования контроллера, установленное на компьютере программное обеспечение Mitsubishi.

На рисунке 1.16 изображен ПЛК ALPHA 2, в таблице 1.7 указаны наименования позиций контроллера.



а – общий вид и размеры; б – основные позиции;  
 в – интерфейсы и каналы ввода-вывода  
 Рисунок 1.16 – ПЛК ALPHA 2

Таблица 1.7 – Наименования позиций контроллера

Номер позиции	Наименование позиции
1	Монтажное отверстие
2	Контактные клеммы подключения питания
3	Контактные клеммы подключения входных цепей
4	Монтажный винт для присоединения корпуса расширителя или расширительного модуля
5	Корпус расширителя или расширительный модуль
6	Канавка для установки рельса в стандарте DIN
7	Монтажные зажимы для установки рельса в стандарте DIN
8	Выходные контактные клеммы
9	Операционные клавиши
10	Жидкокристаллический дисплей
11	Крышка порта связи для программирования

Для установки программы используйте файл Setup из папки ALPCSWIN\_DemoVersion. Во время установки можно выбрать язык (русский). После установки потребуется перезагрузка.

Для записи программы в контроллер нужно запустить программу Mitsubishi Alpha Programming (рисунок 1.17). В настройках указать используемый COM-порт, предварительно соединив ПК и контроллер кабелем (рисунок 1.18), открыть в программе файл прошивки контроллера \*.vls.

Желательно иметь монитор с разрешением не ниже 1024×768 для более удобного просмотра окна программы.



Рисунок 1.17 – Панель инструментов интерфейса программы Mitsubishi Alpha Programming



Рисунок 1.18 – Кабель для связи и программирования ПЛК ALPHA 2 с использованием компьютера

После настройки произведите запись в контроллер, в появившемся окне укажите «Включить режим мониторинга». Запуск и остановку контроллера можно производить кнопками ПУСК и СТОП. Для проверки работоспособности прошивки можно воспользоваться функцией «Симуляция». При этом моделируется работа контроллера и изменяется значение промежуточных переменных.

Для подключения ПЛК необходимо сделать следующее.

1. Изучить руководства по программной и аппаратной части ПЛК ALPHA 2. Ознакомиться с составом оборудования ПЛК.
2. Установить программное обеспечение Mitsubishi Alpha Programming на компьютер.
3. Запустить приложение Mitsubishi Alpha Programming на компьютере.
4. Присоединить интерфейсный кабель к ПЛК, для этого может понадобиться извлечь крышку порта связи ПЛК.
5. Открыть имеющийся проект или пример программы для ПЛК ALPHA 2 в Mitsubishi Alpha Programming.

6. Убедиться, что на панели стенда отсутствуют переключки, все тумблеры (SA2, SA3, SA5, SA6, SA7, SA8) находятся в нижнем положении («Выключено»). Если необходимо, использовать измерительные приборы на панели стенда: должны быть включены тумблеры SA10, SA11.

7. Включить питание стенда (три автоматических выключателя «Сеть»).

8. Записать программу в ПЛК. Запустить моделирование и мониторинг.

9. В окне программы, изменяя состояние входных каналов (1...8 в зависимости от конкретной программы), зафиксировать смену состояний выходов (out).

10. Произвести остановку (СТОП) контроллера.

11. Выключить питание (три автоматических выключателя «Сеть»).

12. Собрать схему, как показано на рисунке 1.19.

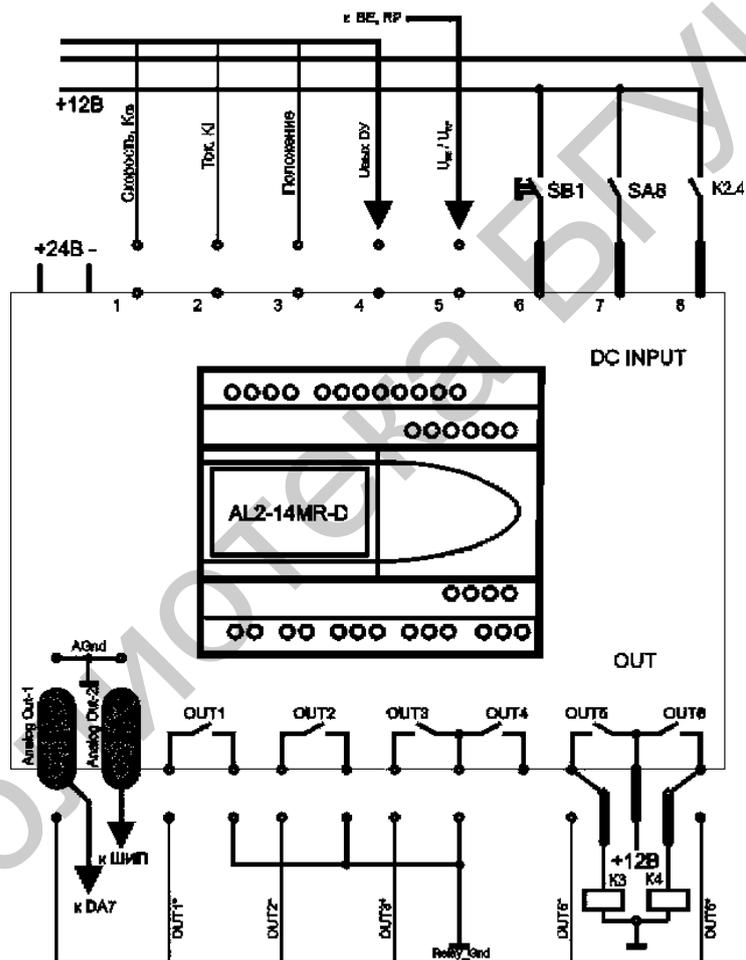


Рисунок 1.19 – Схема подключения ПЛК

13. Убедиться, что схема собрана правильно и на панели стенда присутствуют только необходимые для проведения данного опыта переключки, все тумблеры (SA2, SA3, SA5, SA6, SA7, SA8) находятся в нижнем положении («Выключено»). Если необходимо, использовать измерительные приборы на панели стенда, должны быть включены тумблеры SA10, SA11. Включить питание стенда (три автоматических выключателя «Сеть»).

14. Записать программу в ПЛК. Запустить функционирование и мониторинг.

15. Изменяя состояние входных каналов (6 и 7 в зависимости от конкретной программы) кнопкой или тумблером, зафиксировать смену состояний выходов (out5, out6).

16. Произвести остановку (СТОП) контроллера.

17. Внести изменения в ранее открытую программу, с целью задания нового алгоритма работы.

18. Записать программу в ПЛК. Запустить функционирование и мониторинг.

19. Повторить пункт 8, удостовериться в правильности работы новой программы.

20. Произвести остановку (СТОП) контроллера.

21. Выключить питание (три автоматических выключателя «Сеть»).

22. Разобрать схему.

### **1.10 Контрольные вопросы**

1. Какие основные элементы рабочего экрана в среде AL-PCS/WIN-E вы знаете?

2. Что такое функциональная блок-схема?

3. В каких режимах может вестись работа в AL-PCS/WIN-E?

4. Из каких разделов состоит база функциональных блок-схем (FBD)?

5. Каков порядок создания функциональной блок-схемы (FBD)?

## Практическое занятие №2

### Методы программного управления техническими системами

#### Цель работы:

- 1) изучить основы программного управления техническими системами на базе методов пропорционально-дифференциально-интегрального управления;
- 2) научиться выполнять обоснованный выбор коэффициентов для ПИД-управления;
- 3) освоить программирование программно-управляемых технических систем на основе промышленных контроллеров ALPHA 2.

#### 2.1 Управление с разомкнутой петлей обратной связи

Простейший метод управления механизмами – *управление с разомкнутой петлей обратной связи (ОС)*. Термин «разомкнутая петля» подразумевает, что никакого сигнала обратной связи от управляемого устройства не поступает. Нет индикации того, что устройство выполнило именно то, что ему приказали делать. Примером может служить двигатель виброзвонка в пейджере или сотовом телефоне. Ни пользователь, ни прибор не могут гарантировать, что скорость двигателя не изменилась на 10...20 % от номинальной. Микропроцессор (микроконтроллер) посылает только сигнал включения/выключения двигателю, не имея никакой информации о его скорости. Реальная скорость двигателя будет зависеть от трения на валу, напряжения аккумулятора и состояния щеток. В отличие от приведенного примера, когда скорость двигателя не так уж и важна, большинство микропроцессорных систем управления имеют средства измерения параметров управляемых систем. Для этого необходимо введение цепи ОС от управляемого устройства к микропроцессору.

#### 2.2 Отрицательная обратная связь при управлении системами

На рисунке 2.1 показана простая система управления – *операционный усилитель (ОУ)*. Операционные усилители обладают очень высоким коэффициентом усиления, так как соединением выхода ОУ с инвертирующим входом мы вводим *отрицательную обратную связь (ООС)*. ОУ усиливает разность напряжений между *инвертирующим* и *неинвертирующим входами*. Скажем, на входе и выходе установлено напряжение 2 В. Разность между входом и выходом составит 0 В, тогда разность между инвертирующим и неинвертирующим входами также равна 0 В.

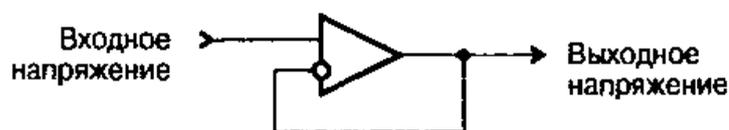


Рисунок 2.1 – Операционный усилитель

Допустим, уровень напряжения на входе резко изменился с 2 до 2,1 В. Тогда появится разница между двумя **дифференциальными входами** – неинвертирующим (2,1 В) и инвертирующим (2 В). Разница в 0,1 В усиливается ОУ до тех пор, пока напряжение на выходе не достигнет 2,1 В, разница между входами снова станет 0 В, и выходное напряжение останется на уровне 2,1В.

Если колебания температуры и дрейф выходных транзисторов ОУ немного изменят характеристики усилителя, выходное напряжение может несколько сместиться. Но как только это произойдет, ОУ усилит разницу между входными напряжениями и снова стабилизирует выходное напряжение. При этом **коэффициент усиления идеального ОУ** измеряется обычно очень большой величиной, однако на практике существуют частотные и другие ограничения для **коэффициента усиления реального ОУ**.

### 2.3 Микропроцессорные системы управления

Микропроцессорные системы управления могут работать по такому же принципу, как и ОУ. МП-система управляет каким-либо прибором (например, двигателем или нагревательным элементом), устанавливающим некоторую физическую величину (позицию или температуру объекта). Секрет, конечно, кроется в функции усиления. В отличие от простого ОУ цифровая система управления может воспроизвести на выходе более сложную функцию. Микропроцессор может сформировать сигнал, являющийся не только функцией входа и выхода, но и предыдущих состояний входа и выхода, коэффициентов изменений, состояния нагрузки и т. п.

Микропроцессорная система управления в отличие от аналоговой всегда является дискретной системой. Это значит, что микропроцессор производит измерения сигнала с датчиков через определенные интервалы. Любые изменения, происходящие между интервалами измерения, теряются. Частота дискретизации должна быть достаточно высокой для гарантии того, что состояние управляемого прибора не изменится значительно за время интервала дискретизации. Интервал дискретизации, естественно, зависит от управляемого прибора и может варьироваться от минут до долей микросекунд.

На рисунке 2.2 показана простая **микропроцессорная система управления**. Микропроцессор включает и выключает МОП-транзистор, управляющий нагревательным прибором. Для измерения температуры применяется терморезистор. Микропроцессор считывает сигнал, пропорциональный температуре, и решает, включить или выключить нагревательный элемент. Величина устанавливаемой температуры подается на «Вход» системы (см. рисунок 2.2).

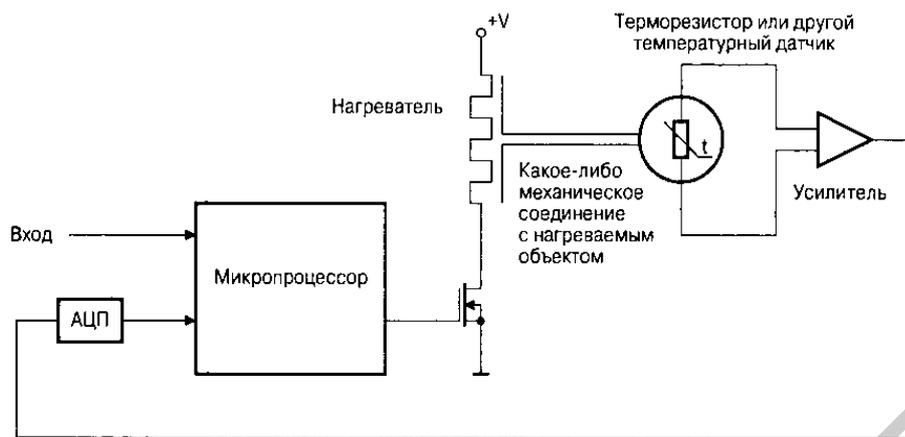


Рисунок 2.2 – Простая микропроцессорная система управления

## 2.4 Система релейного управления

Рассмотрим простейшую *систему релейного управления* (on-off), иногда называемую еще *системой двухпозиционного управления* (bang-bang control). Допустим, микропроцессор считывает температуру датчика. Если температура ниже заданной (определенной температурной) точки, нагреватель включается, если выше – выключается. На рисунке 2.3 показана аналогичная система на основе компаратора.

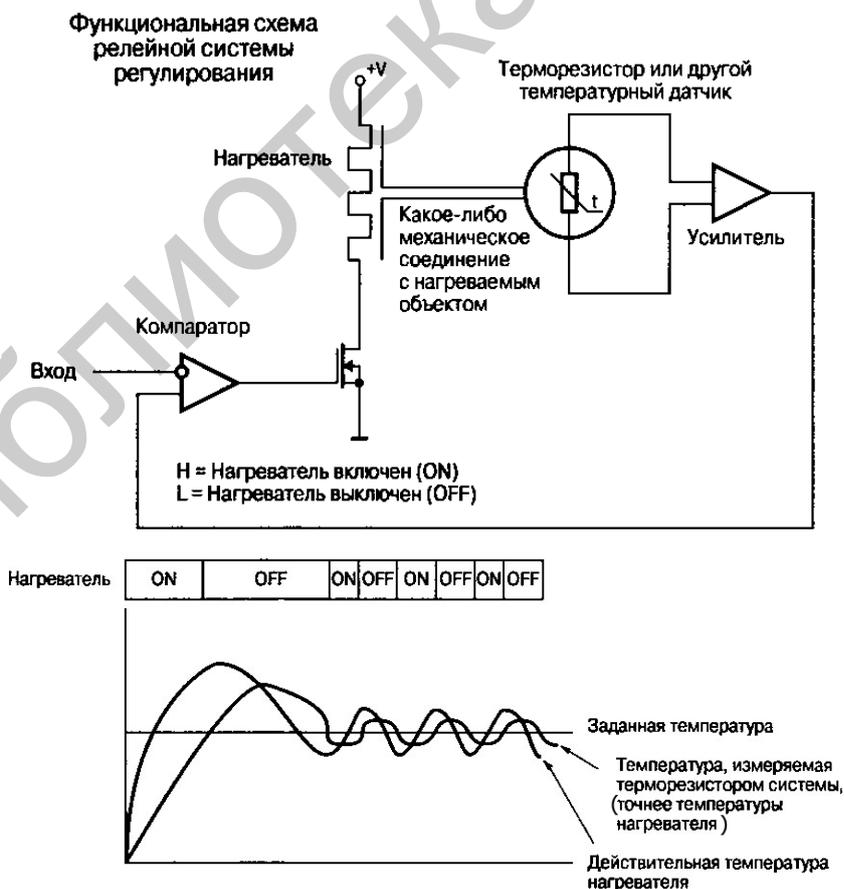


Рисунок 2.3 – Система релейного управления

На данном рисунке также приведена характерная осциллограмма работы системы. Когда система включается, нагреватель холодный. Микропроцессор держит нагреватель во включенном состоянии до тех пор, пока температура терморезистора не достигнет заданного уровня установки. Тогда микропроцессор отключает нагреватель. При понижении температуры ниже заданной нагреватель снова включается. В результате температура объекта колеблется около заданной величины.

На рисунке 2.3 показаны температуры нагревателя и терморезистора. Как можно заметить, они не совпадают ни по времени, ни по амплитуде. При первом разогреве температура нагревателя поднимается на значительную величину, а затем колеблется в широком диапазоне.

Основные причины значительных колебаний температуры нагревателя здесь следующие:

1) недостаточная связь между нагревателем и объектом: температура нагревателя должна быть выше, чтобы передать необходимую энергию для нагрева объекта;

2) процесс нагрева объекта не может произойти моментально, поскольку объект обладает определенной массой и, следовательно, инерционностью;

3) существует задержка между временем достижения заданной температуры объектом и терморезистором, которая обусловлена плохим контактом терморезистора с объектом и конечной теплоемкостью терморезистора;

4) существует задержка между временем включения нагревателя и моментом действительного достижения требуемой температуры; когда же нагреватель выключается, опять же требуется время на остывание.

Рассмотрим пример системы управления, аналогичной представленной на рисунке 2.3. Допустим, цепь управления нагревателем может передавать значительную энергию объекту – много большую, чем необходимо. Нагреваемый объект обладает небольшой массой, примерно как у самого нагревателя, и, соответственно, очень быстро нагревается. Изменения температуры объекта в действительности будут происходить даже быстрее, чем изменения температуры терморезистора. В итоге, когда нагреватель включится, перегрев будет происходить за несколько секунд, а затем секунд 10...20 нагреватель будет выключен до тех пор, пока температура объекта не спадет до требуемого уровня. Таким образом, колебания температуры системы около заданной величины будут слишком высокими. Пример со столь невыгодными условиями приведен специально, чтобы продемонстрировать, что релейная система будет устойчивей работать при выполнении следующих требований:

1) управляемый объект не должен быстро реагировать на управляющий сигнал;

2) измеряющий датчик, напротив, должен очень быстро воспроизводить все изменения системы.

В примере с обогревателем эти требования выглядели бы следующим образом: **теплоемкость объекта** должна превышать **теплоемкость нагревательного прибора**, а терморезистор должен быть плотно присоединен к объекту, чтобы иметь с ним надежный **тепловой контакт** и быстро реагировать на изменения температуры. Местоположение терморезистора может оказать значительное влияние на работу системы.

В описанной системе терморезистор находился между нагревателем и объектом (из-за пространственных ограничений). Следовательно, показания терморезистора соответствовали некоему среднему значению между двумя температурами. Если бы объект имел значительную массу и теплоемкость, температура его никогда не смогла бы достичь требуемого уровня при таком включении.

*Обогреватель в квартире или доме – хороший пример релейной системы. Объем воздуха в доме довольно большой и долго нагревается, то есть обогреватель не может быстро изменить температуру помещения. Термостат – медленный прибор по сравнению с микропроцессором, но довольно точно отражает изменения температуры воздуха в доме.*

*Недостаток такой системы в больших по сравнению с обогревателем размерах дома. Если обогреватель выключить на весь день, в то время как температура на улице падает, то понадобится значительное время для того, чтобы вернуть температуру в доме до комфортного уровня. Нет никакой возможности для быстрой передачи энергии системе. Правда, можно было бы купить обогреватель больше в несколько раз, который быстро нагрел бы дом, но тогда не избежать перегревов и значительных колебаний температуры.*

В некоторых **системах релейного управления** есть так называемая **мертвая зона** (dead band), в пределах которой выходной сигнал не изменяется. Это позволяет предотвратить чрезмерные переключения системы.

Например, система терморегулирования могла бы иметь мертвую зону в  $1^\circ$ , то есть выходной сигнал не менялся бы при температуре объекта внутри данного диапазона. Если нагреватель включен, то он не выключится, пока температура объекта не превысит установленное значение на  $1^\circ$ . Также нагреватель не включится, пока температура не опустится на  $1^\circ$  ниже заданной величины. В некоторых системах мертвая зона составляет даже часть общего физического принципа работы. Например, термостат в некоторых домах несколько отстает от изменения температуры, поэтому термостат будет чувствовать продолжающееся повышение температуры, после того как необходимый уровень достигнут, и нагреватель выключился. На самом же деле термостат отображает действительную температуру окружающей среды.

В примере с нагревателем (см. рисунок 2.3) на осциллограмме виден значительный температурный перегрев. Однако не во всякой системе перегрев будет иметь место. **Температурный перегрев в системе релейного управления** обычно результат некоторой инерции системы. В рассмотренном примере нагреватель продолжает нагревать объект по инерции, после выключения питания. К тому же, если нагреватель достаточно большой по отношению к

объекту, и к нагревателю приложена значительная мощность, время нагрева будет меньше, чем время остывания. Подобные рассуждения можно также применить, например, к двигателям, перемещающим нагрузку с большим моментом инерции.

## 2.5 Система пропорционального управления

На следующем уровне по сложности (после описанной выше релейной системы управления) находится **система пропорционального управления**. Принцип пропорционального управления базируется на изменении управляющего сигнала в зависимости от разности между истинным сигналом и заданной величиной. Разница между истинной и заданной величинами называется «ошибкой». Формула вычисления истинной величины выходного сигнала (*Output*) пропорционального управления следующая:

$$Output = G \cdot e, \quad (2.1)$$

где  $G$  – это усиление,  $e$  – **величина ошибки** (разница между заданной и истинной величинами) в **системе пропорционального управления**.

Система с пропорциональным управлением больше похожа на ОУ с ограниченным усилением, чем на компаратор (систему с очень большим усилением). Реальная система управления может представлять собой аналоговую систему под управлением микроконтроллера (цифроаналоговый преобразователь (ЦАП) и усилитель) либо систему с широтно-импульсным модулятором (ШИМ).

На рисунке 2.4 приведен пример системы пропорционального управления. Нагревательный элемент включен на 100 %, пока объект холодный. Когда температура объекта достигает заданной величины, нагрев уменьшается, поскольку разница между заданной и истинной (измеряемой) температурами становится меньше. Как можно заметить, система пропорционального управления обладает меньшими величинами перегрева и колебаний от заданного значения. На рисунке 2.4 показаны колебания температуры примерно в 2 раза меньшие, чем для системы с релейным управлением (см. рисунок 2.3). Действительное отличие между системами будет зависеть от конкретного случая.

Другое преимущество пропорционального управления заключается в том, что можно подстроить управляющий сигнал в зависимости от управляемого объекта. Например, если вы хотите нагреть жидкость, текущую по трубе, то при возрастании скорости потока понадобится пропорционально увеличить коэффициент усиления, чтобы добиться нагревания потока.

Для некоторых систем выходной сигнал можно представить, как  $G \cdot e$ , как показано ранее. Но для многих систем, в том числе и для изображенной на рисунке 2.4, формула пропорционального управления будет выглядеть следующим образом:

$$Output = (G \cdot e) + M, \quad (2.2)$$

где  $M$  – это величина смещения в системе пропорционального управления, необходимая для поддержания управляемой величины на требуемом уровне.

В примере с нагревателем система в соответствии с данным уравнением уменьшит мощность, как только температура приблизится к требуемой величине ( $G \cdot e$  приблизится к нулю), и в конечном итоге температура никогда не достигнет требуемого предела. Обычно, если  $M$  составляет 50 % от мощности нагревателя, система может достичь требуемой величины. В формуле (2.2) пропорционального управления величина  $G \cdot e$  складывается с величиной постоянного смещения  $M$ . Пока система холодная, величина  $G \cdot e$  будет слишком большой, и нагреватель будет работать на все 100 %. Если сумма  $(G \cdot e) + M$  смещения станет больше 100 %, выходная мощность все равно останется на уровне 100 %, поскольку большую мощность система не может отдать. При достижении нагревателем заданной температуры величина  $G \cdot e$  станет меньше, и мощность нагревателя начнет уменьшаться вплоть до 50 % от смещения. Если температура превысит заданную, то  $G \cdot e$  станет отрицательной, и сумма  $(G \cdot e) + M$  станет меньше чем 50 %, уменьшая, таким образом, мощность нагревателя. Если сумма в формуле (2.2) станет ниже нуля, выходное напряжение отключится до тех пор, пока сумма не станет снова положительной. Пример возможности реализации отрицательного знака на выходе – система с нагревателем и охладителем.

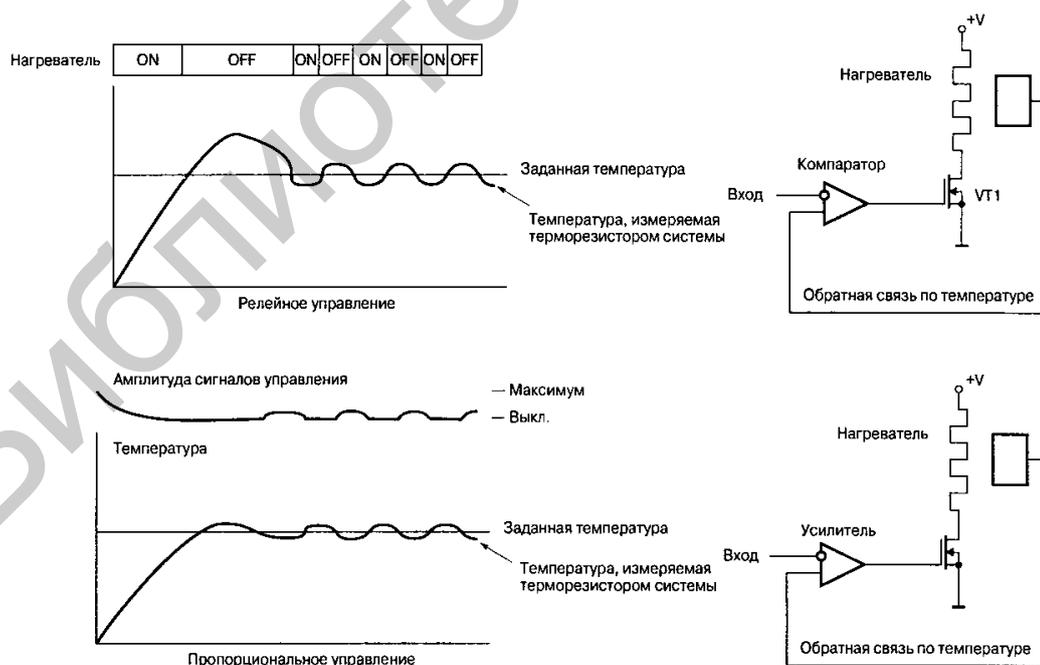


Рисунок 2.4 – Система пропорционального управления

Заметим, что  $M$  может меняться с изменениями характеристик системы. Представьте, что вы нагреваете металлические блоки. Для нагрева небольших блоков достаточно будет установить  $M$  на уровне 20 %, тогда как массивные изделия потребуют довести  $M$  до 80 % номинальной мощности нагревательного элемента.

Проектирование системы пропорционального управления – более сложная задача, чем разработка системы релейного управления.

В релейной системе без перегрева не обойтись, что неизбежно приведет к колебаниям на выходе и избыточной мощности на нагрев. При изменении нагрузки величина и перегрева, и колебаний будет меняться в зависимости от того, насколько возможно будет добавить дополнительную энергию в систему. В системе пропорционального управления необходимо подстроить коэффициент усиления и смещение в соответствии с объектом. Задав слишком большое усиление, получим систему с релейным регулированием. Снизив усиление, можно никогда не достичь заданной температуры. Хуже того, если, например, некоторая нагревательная система с пропорциональным регулированием работает в лабораторных условиях, то на улице в холодное время года может и отказать. Лучше всего системы с пропорциональным регулированием работают в режиме, когда нагрузка постоянна и хорошо известна. Примером известной нагрузки может служить нагреватель пластинки постоянной формы и размера или нагреватель жидкости, поток которой может меняться, но в известных пределах.

Надо заметить, что некоторые системы пропорционального управления не нуждаются в задании смещения. Например, управление двигателем, сдвигающим его ось в определенное положение, не нуждается в смещении. Вместо этого выходной параметр двигателя, определяющий скорость, выражался бы произведением усиления на ошибку. В данном случае ошибка управления – это разность между действительным и заданным положением оси двигателя, и в момент, когда ошибка управления становится равной нулю, двигатель останавливается. С другой стороны, система, в которой двигатель должен вращаться с постоянной скоростью, потребует задания смещения, поскольку при условии, когда ошибка управления равна нулю (действительная скорость вращения равна требуемой), к двигателю должна подводиться необходимая мощность.

Во многих случаях трудно разработать систему пропорционального управления, устанавливающую выходной уровень без колебаний. В большинстве случаев окончательная величина системы (температура, скорость и т. п.) – несколько ниже заданной. Действительно достигнутая величина зависит от усиления, смещения ( $M$ ) и нагрузки.

Проблема системы пропорционального управления в том, что такая система подгоняет управляющий сигнал в зависимости от разности между заданной и измеряемой величинами. Нет механизма подстройки управляющего сигнала к условиям, о которых микропроцессор не имеет представления. Например, что произойдет при нагреве пластины металла, масса которой в 2

раза больше нормы, или если в нагреватель вместо медной попадает алюминиевая пластина? Можно, конечно, добавить датчик веса в систему, однако, что если окажется изделие той же массы, но другой формы? Ведь длинная тонкая пластина будет нагреваться по-другому, нежели образец, покрывающий всю площадь нагревателя.

Подобные проблемы существуют и в других системах управления. Автомобильный круиз-контроль, например, должен учитывать влияние таких внешних воздействий, как встречный ветер, крутые спуски и подъемы и т. п. Система пропорционального управления создала бы много проблем в таких условиях и поэтому в круиз-контроле не применяется.

## 2.6 Пропорционально-интегрально-дифференциальное управление (ПИД-управление)

Система, выполняющая пропорциональное, интегрирующее и дифференциальное управление, носит название *системы пропорционально-интегрально-дифференциального управления*, или сокращенно *системы ПИД-управления* (PID – Proportional, Integral, Derivative (control)). Система ПИД-управления добавляет в систему дополнительный вход, несущий информацию о предыдущем состоянии системы. В системе круиз-контроля, например, вместо изменения скорости за счет вычисления разности между требуемой и заданной скоростью автомобиля (пропорциональное управление), управление скоростью может осуществляться в зависимости от того, как автомобиль отреагировал на предыдущее воздействие. Ускорение произошло не так быстро, как ожидалось? Это признак того, что автомобиль движется в гору или усилился встречный ветер.

На рисунке 2.5 показана структурная схема системы ПИД-управления. Разница между заданной и измеряемой величинами усиливается. Производная и интеграл усиленной разности суммируются с усиленным сигналом ошибки, формируя сигнал на выходе.

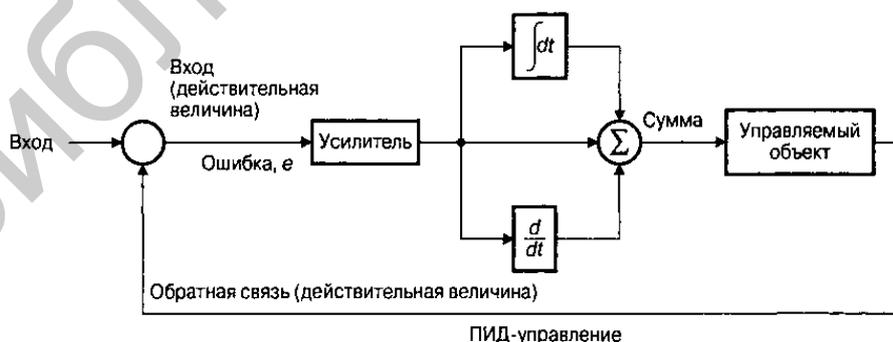


Рисунок 2.5 – Система ПИД-управления

Общая формула для ПИД-систем выглядит следующим образом:

$$Output = G \cdot \left( e + I \cdot \int e dt + D \cdot \frac{de}{dt} \right), \quad (2.3)$$

где  $G$  – усиление (Gain);  $e$  – ошибка управления, то есть разность между требуемой и действительной величинами (error);  $I$  – прибавляемая величина интеграла ошибки (Integral);  $D$  – прибавляемая величина производной ошибки (Derivative).

Если  $I$  и  $D$  равны нулю, получаем формулу пропорционального управления:  $Output = G \cdot e$ . Если  $I$  и  $D$  равны нулю, а  $G$  настолько велико, что выходной сигнал доходит до насыщения, получаем релейную систему управления. Как и в системах с пропорциональным регулированием, в ПИД-системах иногда требуется задать некоторое смещение, которое может в данном случае как прибавляться, так и вычитаться.

От системы пропорционального управления система ПИД-управления отличается дополнительными вычислениями интеграла и производной. Это уже зависимые от времени параметры: интеграл вычисляется за некоторый период времени, а производная – это разность между состояниями в соседних временных интервалах.

Любая система характеризуется некоторой инерцией. Когда вы включаете устройство для нагрева чего-либо, нагреватель обязательно будет горячее объекта. При выключении нагреватель не остынет мгновенно. Температура его будет снижаться плавно. В то время как нагреватель охлаждается, температура объекта может продолжать расти. На рисунке 2.6 показан такой случай. Разность температур между нагревателем и объектом, а также скоростей нагревания и охлаждения зависит от массы объекта, мощности нагревателя, контакта между нагревателем и объектом, формы объекта и т. п.

На рисунке 2.6 показаны кривые управления *легкой и тяжелой нагрузками* одной и той же *системы*. При нагреве металлических блоков, например под тяжелой нагрузкой, может подразумеваться блок с большей массой либо сделанный из железа вместо алюминия. Как видно из графиков, более массивный блок нагревается и остывает дольше, то есть обладает большей инерцией. Если рассматривать управление скоростью автомобиля, тяжелой нагрузкой будет ускорение при движении на крутой подъем, а легкой нагрузкой – ускорение при крутом спуске.

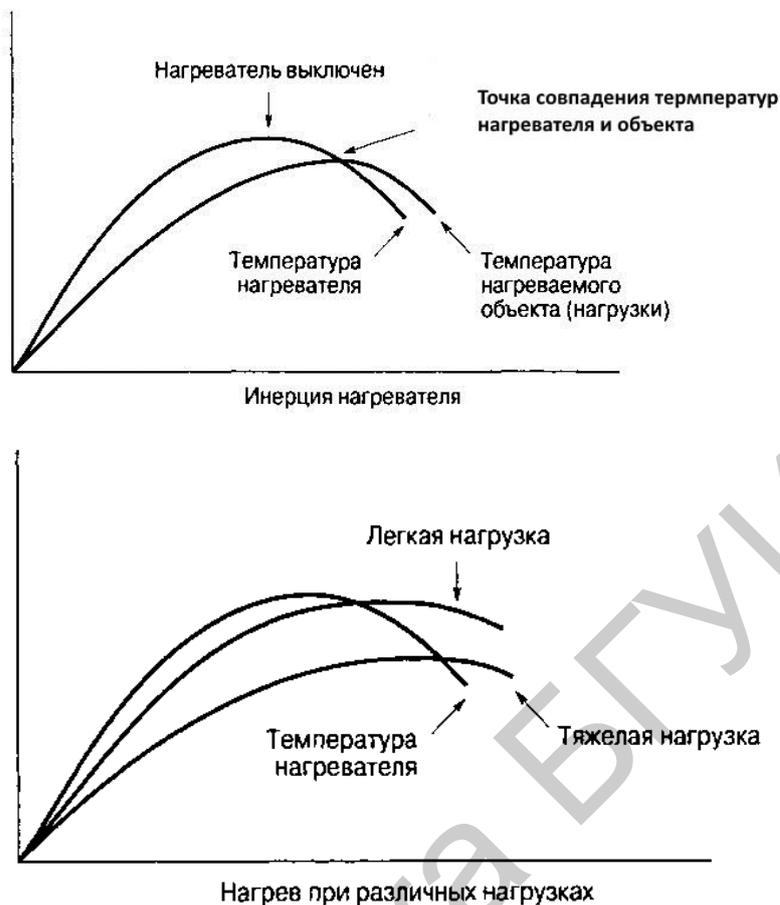


Рисунок 2.6 – Инерция в системе управления

Допустим, нагреватель находится под управлением системы пропорционального управления. Поскольку мощность нагревателя определяется только разностью между заданной и действительной температурами, управляющий сигнал будет одинаковым как для легкой, так и для тяжелой нагрузки. Это значит, что легкая нагрузка перегреется сильнее. Следовательно, при заданной температуре система будет совершать колебания большей амплитуды. Точность установленной температуры для легкой нагрузки окажется гораздо ниже, чем для тяжелой нагрузки.

### 2.6.1 Вклад операции дифференцирования в алгоритм управления

Добавление вычисления производной (дифференцирования) в алгоритм управления позволяет улучшить некоторые параметры. Производная ошибки в данном случае отражает скорость изменения ошибки управления. Если системе управления известна мощность нагревателя, то скорость изменения ошибки управления дает информацию о природе нагрузки.

Графически производная некоторой функции в точке представляет собой наклон кривой этой функции, в нашем случае – наклон кривой изменения

ошибки. Если величина ошибки управления уменьшается, производная будет отрицательной, если увеличивается – производная станет положительной. Если величина ошибки управления постоянна и не изменяется, производная будет равна нулю. Заметим, что даже в случае большой, но постоянной величины ошибки управления, производная будет равна нулю. Графики изменения температуры нагревателя/объекта, величины ошибки управления и ее производной показаны на рисунке 2.7.

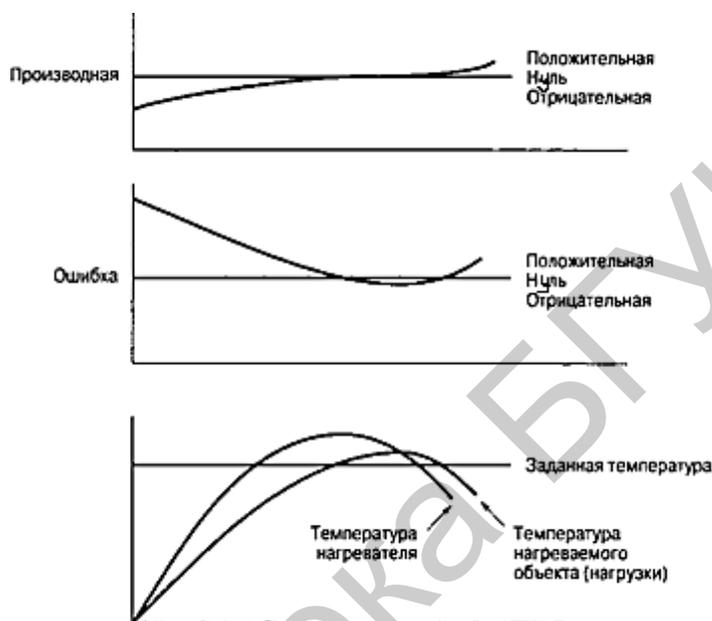


Рисунок 2.7 – Производная составляющая в системе управления

Если уменьшить коэффициент усиления, затем добавить производную величины  $G \cdot e$ , система будет лучше отслеживать вариации «нагрузки».

Если «нагрузка» нагревается довольно быстро, производная будет иметь большую отрицательную величину, и выходной сигнал  $\left(G \cdot e + D \cdot \frac{de}{dt}\right)$  будет меньше. Меньший выходной сигнал обозначает меньший нагрев: так объект будет нагреваться медленнее. В случае тяжелой нагрузки производная будет иметь небольшую отрицательную величину, на нагреватель пойдет большая мощность, и нагрев будет происходить быстрее.

Когда температура объекта приблизится к заданной величине, величина  $G \cdot e$  уменьшится. Снижение мощности нагревателя также приведет к уменьшению производной, в результате установление заданной температуры произойдет с меньшим перегревом. При превышении температуры объекта заданной величины производная поменяет знак, как и величина ошибки. Это увеличит положительную величину в выражении коэффициента усиления. Чем быстрее будет остывать объект, тем больше будет производная ошибки, что снизит пульсации выходной мощности.

На рисунке 2.8 показана работа системы регулирования температуры объекта с пропорциональным усилением и учетом производной ошибки управления. Заметен небольшой перегрев, за которым следуют пульсации около заданной температуры. В зависимости от параметров системы перегрев, а также пульсации можно уменьшить или вообще свести к нулю.

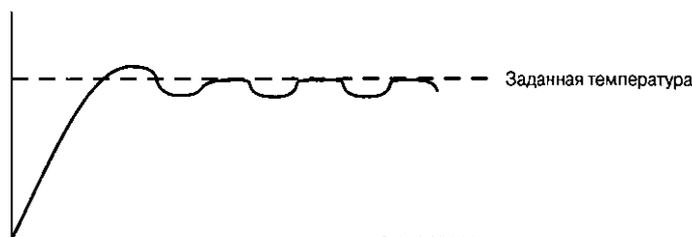


Рисунок 2.8 – Пропорциональное усиление и производная ошибки управления

Как видно из графика, результирующая температура оказывается немного ниже заданной. Это происходит за счет того, что усиление системы недостаточно высоко. Когда температура объекта приближается к заданной величине, наклон изменения ошибки снижается, производная стремится к нулю. На рисунке 2.9 показан график выходного сигнала системы с усилением и учетом производной, в которой окончательная ошибка управления составляет малую постоянную величину.

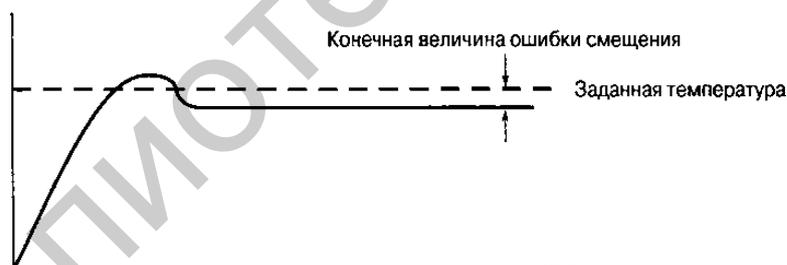


Рисунок 2.9 – Ошибка управления

### 2.6.2 Вклад операции интегрирования в алгоритм управления

Один из методов корректировки окончательной небольшой ошибки системы управления – введение вычисления интеграла ошибки. Графически интеграл представляет собой площадь области под кривой. В нашем случае интеграл – это сумма величин ошибок за определенный период.

На рисунке 2.10 показан график интеграла ошибки. Заметим, что в этом примере интеграл не становится отрицательной величиной, даже в случае отрицательных ошибок. Если же величина ошибки будет отрицательной продолжительное время, то интеграл в итоге также станет отрицательным.

На рисунке 2.11 показан график температур рассмотренного ранее нагревателя при пропорционально-разностном управлении и с учетом интеграла постоянной ошибки.

Когда система стабилизируется с небольшим смещением, величина интеграла ошибки начинает расти, поскольку ошибка управления накапливается (в данном случае, температура немного ниже заданной, ошибка управления положительная, интеграл растет в положительном направлении). В итоге величина интеграла становится достаточно большой для смещения выходной температуры до необходимого уровня.

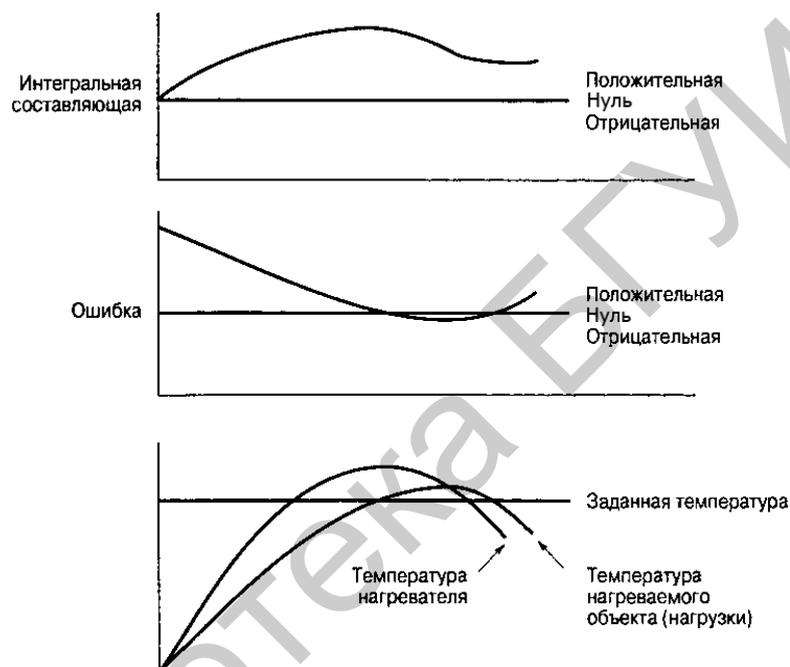


Рисунок 2.10 – Интегральная составляющая в системе управления

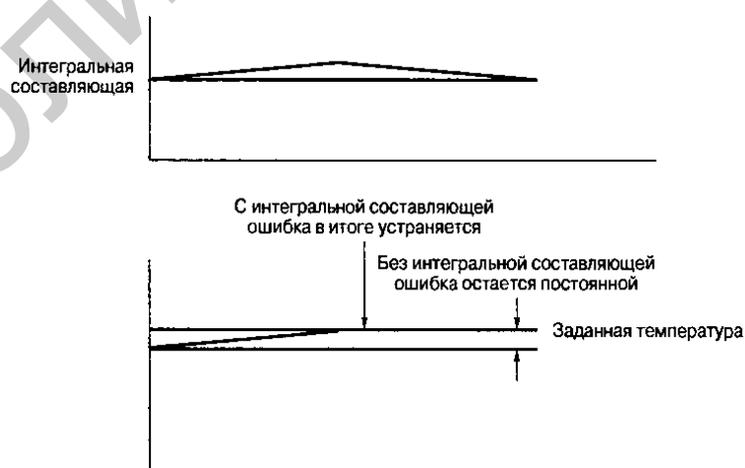


Рисунок 2.11 – Управление с учетом интегральной составляющей ошибки управления

Так, к примеру, в круиз-контроле система с пропорционально-разностным управлением могла установить скорость автомобиля на уровне 99,2 км/ч (62 миль/ч) при заданной скорости 104 км/ч (65 миль/ч). Если автомобиль достаточно время движется со скоростью 99,2 км/ч (62 миль/ч), то интеграл ошибки сможет возрасти до необходимой коррекции выходной величины.

### 2.6.3 Система управления с критическим затуханием

Пропорциональная часть системы ПИД-управления заставляет выходной сигнал следовать за входным (установить заданную величину). Вычисление производной обеспечивает реакцию выходного сигнала на быстрые изменения входного в целях компенсации влияния «нагрузки». Интегрирование компенсирует медленно изменяющиеся погрешности.

Во всех перечисленных ранее примерах присутствует превышение выходного параметра (перегрев) и некоторые колебания около положения равновесия. Такие осциллограммы характерны для систем с малым затуханием. На рисунке 2.12 показана реакция системы с критическим затуханием. Здесь реакция системы резко возрастает почти до заданного уровня, однако не превышает его и не совершает колебаний, когда уровень достигнут. Во многих системах небольшое превышение выходного сигнала над заданным – это плата за высокое быстродействие. В других случаях используется критический режим, позволяющий установить заданную величину без перегрузок и колебаний. Примером может служить система охлаждения для поддержания биологических образцов вблизи температуры замерзания, но не позволяющая допустить переохлаждения, поскольку биологические объекты разрушаются при замерзании.

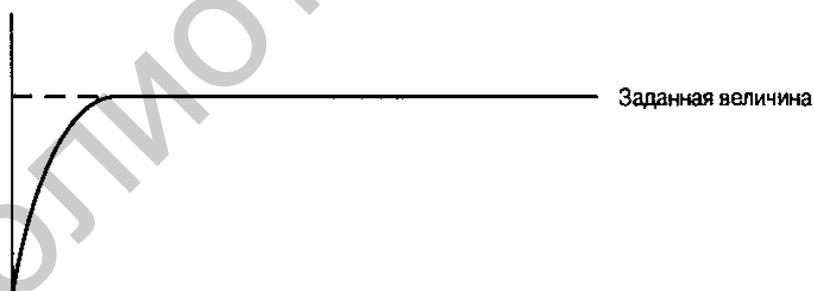


Рисунок 2.12 – Система управления с критическим затуханием

В классической теории управления усиление интегральной и дифференциальной оценок является безразмерной величиной, как и коэффициент пропорционального усиления. В микропроцессорных системах сигналы дискретизируются с заданным интервалом, и может оказаться так, что установить желаемую величину интервала интегрирования или дифференцирования невозможно. Однако, используя определенные коэффициенты для сигналов оценок интеграла и производной, возможно подвести эти величины к требуемым временным параметрам.

## 2.6.4 Способы применения системы ПИД-управления

Хотя система ПИД-управления может работать с различными нагрузками, она требует определенной подстройки. Подстройка – процесс выбора параметров (коэффициентов) всех трех величин, то есть сколько интегральных и дифференциальных оценок может быть добавлено к усилению системы ( $G \cdot e$ ) и какой должна быть величина  $G$ . Существует много путей решения данной задачи, например, *метод Циглера/Николса* (Zigler/Nichols). Первоначальная сложность настройки в том, что подстраиваемый параметр влияет на другие два. Следующая сложность в том, что результаты моделирования обычно резко отличаются от работы реальной системы.

В общем процедура подстройки системы ПИД-управления заключается в том, чтобы установить усиление  $G$  достаточно большим для обеспечения высокой скорости работы системы.

Затем величина производной  $D$  устанавливается достаточно большой для уменьшения возможного избыточного усиления и колебаний. Наконец, интегральный коэффициент  $I$  также делается большим для устранения ошибки установившегося режима.

Например, по методу Циглера/Николса производятся следующие шаги:

1. Отключаем цепи интегрирующего и дифференцирующего сигналов, что превращает систему управления в систему с пропорциональным регулированием.

2. Увеличиваем усиление до тех пор, пока на выходе не установятся незначительные либо затухающие колебания. Назовем такой уровень усиления  $K$ .

3. Измеряем период колебаний  $P$ .

4. Устанавливаем коэффициент пропорционального усиления  $G$ , а также коэффициенты усиления сигналов интеграла и производной  $T_i$ ,  $T_d$  в соответствии со следующими соотношениями:

- если система управления только пропорциональная, то  $gain = 0,5K$ ;
- если система пропорционально-интегрирующая, то  $G = 0,45K$ ,  $T_i = 1,2/P$ ;
- если мы создаем систему ПИД-управления, то  $G = 0,6K$ ,  $T_i = 2/P$ ,  $T_d = P/8$ .

Результат данных вычислений, конечно, потребует дополнительной корректировки для оптимизации работы. Как упоминалось ранее, в результате могут быть получены такие величины интеграла и производной, которые неосуществимы при заданной частоте дискретизации. В этом случае коэффициенты должны быть подобраны экспериментально.

*Подстройка системы с ПИД-регулированием порой приводит к новым проблемам, таким как, например, трудности измерений. В процессе регулирования оценивается способность контроллера поддерживать определенные скорости двигателя, превышение скорости может вывести электронику из строя. В холодильнике, например, этот период времени может составлять как минуты, так и часы. Какое время продлится переходный*

процесс в системе при тех или иных параметрах Циглера/Николса? Подобные проблемы могут превратить разработку устройства в целое исследование.

Реальные системы часто работают не совсем так, как их модели. Проблемы, возникающие в системах с ПИД-регулированием, перечислены ниже.

### 2.6.5 Насыщение

Существует возможность рассчитать такой сигнал на выходе, который реальной электромеханической системой никогда не будет достигнут. Например, если поместить большой холодный металлический блок в рассмотренный ранее нагреватель, система может решить, что для установления необходимой температуры потребуется чрезмерно высокая сила тока. Данная величина силы тока может оказаться за пределами возможностей источника питания и нагревателя либо слишком мощный источник может в таком случае вывести нагреватель из строя.

Другая проблема, связанная с насыщением, заложена в интегральном слагаемом. Если нагреватель будет работать на 100 % мощности, интегральная погрешность будет расти со временем. Так как система не сможет среагировать так быстро, как в ненасыщенном режиме, интегральная погрешность может оказаться слишком большой.

Когда требуемая величина достигнута и слагаемые производной и усиления останавливают управление нагрузкой, интегральное слагаемое будет продолжать заставлять работать систему в том же направлении. Это условие называется отменой первоначального действия (wind up). На рисунке 2.13 показано, как отмена первоначального действия может повлиять на выходной сигнал.

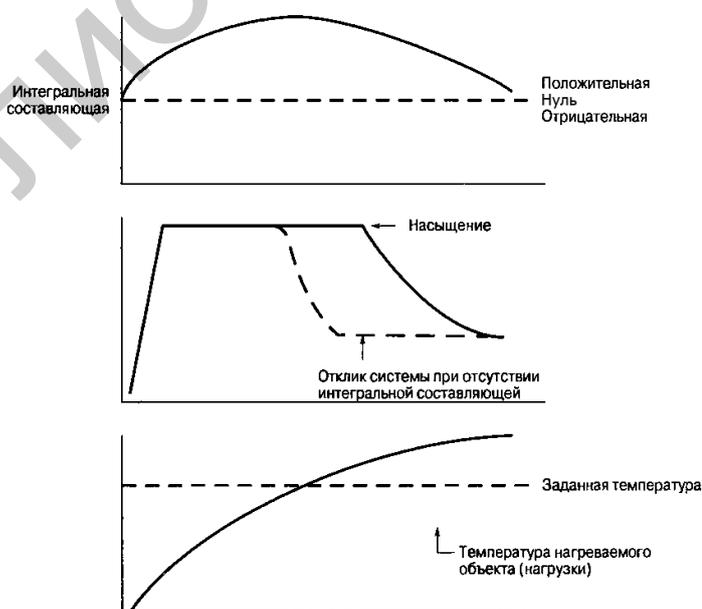


Рисунок 2.13 – Отмена первоначального действия

Датчик, например, масштабированный терморезистор, также может достичь насыщения. Даже если измеряемое значение попадает в температурный диапазон терморезистора, это может привести к насыщению выхода ОУ.

### 2.6.6 Программное обеспечение

Для того чтобы избежать отмены первоначального действия, программное обеспечение должно ограничивать интегральное слагаемое при обнаружении насыщения выходного сигнала или сигнала с датчика. К тому же реальное ПО в отличие от математической модели работает с регистрами конечных размеров. Следует сделать так, чтобы ограниченный размер регистров не привел к сбою системы.

Иногда вычисление интеграла запрещается, если величина погрешности отличается на доли процента от искомой величины. Это позволяет избежать больших интегральных оценок, когда устанавливаемая величина находится вблизи требуемой.

Компонента производной в системе ПИД-управления – это величина изменения погрешности за определенное время. Поскольку в микропроцессорной системе частота дискретизации постоянна, это время обычно равняется интервалу дискретизации (или кратно ему). Тогда производная вычисляется вычитанием двух соседних отсчетов. Вычитание погрешности в момент  $n$  из погрешности в момент  $n + 1$  дает величину изменения в одном временном промежутке. Для снижения шумов программное обеспечение может усреднить несколько вычисленных таким образом значений.

Интеграл вычисляется как сумма погрешностей за определенное время. В реальной микропроцессорной системе интеграл находят как сумму средних величин отдельных отсчетов. И снова необходимо принять во внимание возможность перегрузки и насыщения системы.

### 2.6.7 Временная задержка

Один из параметров, который был упомянут без детального изучения, – временная задержка. Вновь возьмем пример с нагревателем, который рассматривался ранее, и введем несколько временных задержек, включающих:

- время, необходимое нагревателю для установления температуры в соответствии с управляющим сигналом;
- время передачи тепла от нагревателя нагрузке;
- время, необходимое терморезистору для оценки температуры нагрузки.

Перечисленные задержки вносят неточность в систему управления. Время, проходящее от подачи сигнала управления до реакции датчика, называется **мертвым временем**. Если микропроцессор изменяет сигнал из-за того, что объект слишком холодный, нагревателю потребуется некоторое время на разогрев, передачу тепла нагрузке и отклик терморезистора. Тем временем микропроцессор проанализировал терморезистор несколько раз, определил, что температура не в норме и увеличил выходной сигнал. Так, например, в релейной

системе задержка отклика может привести к нежелательным колебаниям выходной величины. В другом случае система управления может чрезмерно скомпенсировать вычисляемую погрешность. Используя ПИД-управление вместо пропорционального, можно уменьшить некоторые из описанных эффектов, как мы уже видели. Однако в некоторых случаях ПИД-управление может только ухудшить ситуацию за счет негативного воздействия интегральной составляющей, например, когда происходит отмена первоначального действия.

Компенсация мертвого времени обычно включает предсказание эффекта от изменения управляющего сигнала и предположение, что эффект произойдет по истечении этого мертвого времени. После измерения реакции системы на некоторое изменение следующее изменение сигнала производится на основе разности между действительной и теоретической величиной. Такой процесс носит название *предсказатель Смита* (Smith Predictor) и был разработан Отто Смитом в 1957 г. Этот процесс включает моделирование работы системы для того, чтобы предсказать отклик.

### 2.6.8 Резкие изменения сигнала

Многие системы испытывают столь неожиданные изменения входных сигналов, что становится нецелесообразно применять пропорциональное или ПИД-управление. Допустим, нагревается резервуар с выплескивающейся жидкостью. Жидкость может резко охлаждать нагреватель, что приведет к сложности настройки системы ПИД-управления. Как предсказать, сколько жидкости может выплеснуться на нагреватель?

Другой пример неожиданного изменения параметров нагрузки – бортовая сеть автомобиля. Если водитель вдруг выключает фары, нагрузка электрической сети резко уменьшается (это называется *сбросом нагрузки*). Напряжение генератора резко поднимается до большой величины, так как генератор пытается обеспечить ту же мощность при пониженном уровне тока. В такой системе ПИД-регулирование не подходит, так как нужен очень быстрый отклик системы. Типичный путь сглаживания подобных выбросов – применение регуляторов с ШИМ и установление на выходе определенной величины до тех пор, пока процесс не стабилизируется. Решение подобных проблем – предварительная оценка величин интеграла и производной сигнала и проектирование такой системы, чтобы эти величины не принимали слишком больших значений.

### 2.6.9 Специальные требования

Многие системы с ПИД-регулированием должны поддерживать специфические входные сигналы. Системе круиз-контроля может понадобиться переходить в ждущий режим, обнуляя интегральную и дифференциальную оценки, когда водитель нажимает на тормоз. Регулятор напряжения бортовой сети автомобиля может работать в различных режимах, с множеством ПИД-

параметров, в широком диапазоне скоростей. У нагревателя должно быть время включенного состояния, необходимое для нагрева различных материалов. Все специфические требования к системе должны быть отражены в ПО, чтобы при некоторых изменениях, например нагреваемого материала, ПИД-управление не вышло бы из-под контроля. Когда параметры системы меняются по какой-либо причине, ПО должно установить соответствующие ПИД-параметры.

## 2.7 Порядок выполнения работы

1. Изучить теоретическую часть практического занятия.
2. Собрать схему, заданную преподавателем.
3. Убедиться, что схема собрана правильно и на панели стенда присутствуют только необходимые для проведения данного опыта переключки, все тумблеры (SA2, SA3, SA5, SA6, SA7, SA8) находятся в нижнем положении («Выключено»). Если необходимо использовать измерительные приборы на панели стенда, должны быть включены тумблеры SA10, SA11.
4. Включить питание стенда (три автоматических выключателя «Сеть»).
5. Запустить приложение Mitsubishi Alpha Programming на компьютере.
6. Присоединить интерфейсный кабель к ПЛК, для этого может понадобиться извлечь крышку порта связи ПЛК.
7. Открыть имеющийся проект или создать программу для ПЛК ALPHA 2 в Mitsubishi Alpha Programming. Контроллер во время исполнения программы должен по команде с переключателя SA8 включать и отключать следящую систему. При этом заданием системы является сигнал рассогласования потенциометров С1 и С3. Исполнительный двигатель – М3.
8. Записать программу в ПЛК. Запустить функционирование и мониторинг.
9. Включить следящую систему переключателем SA8.
10. Установить ручку потенциометра С1 в положение 0.
11. Плавно вращая ручку потенциометра С3, снять зависимости  $URP=f(p)$ ,  $U_{out1}=f(p)$  и  $U_{outA1}=f(p)$  с интервалом  $30^\circ$  на всем обороте. Данные занести в таблицу 2.1.

Таблица 2.1 – Результаты измерений

Угол поворота $\Phi, ^\circ$	Показатели		
	$URP, В$	$U_{out1}, В$	$U_{outA1}, В$
0			
30			
60			
90			
...			

12. Установить ручку потенциометра С1 в крайнее правое положение и повторить измерения (пункт 11).

13. Выключить переключатель SA8.

14. Произвести остановку (СТОП) контроллера.

15. Выключить питание (три автоматических выключателя «Сеть»).

16. Разобрать схему.

## 2.8 Контрольные вопросы

1. Какие методы управления в технических системах вы знаете?

2. Каковы особенности управления без обратной связи?

3. Какие преимущества дает использование обратной связи при управлении в технических системах?

4. Каков принцип пропорционального управления?

5. Каковы особенности использования дифференциальной составляющей в ПИД-системах управления?

6. Каковы особенности использования интегральной составляющей в ПИД-системах управления?

## Практическое занятие №3

### Программное управление двигателем постоянного тока

#### Цель работы:

- 1) изучить статические механические и электромеханические характеристики двигателя постоянного тока с независимым возбуждением при работе в замкнутой системе автоматического регулирования (САР) на основе контроллера ALPHA 2;
- 2) освоить методику расчета статических характеристик двигателя постоянного тока с независимым возбуждением при работе в замкнутой САР;
- 3) получить практические навыки по экспериментальному исследованию статических характеристик двигателя постоянного тока с независимым возбуждением на универсальном лабораторном стенде.

#### 3.1 Двигатель постоянного тока

На рисунке 3.1 показано сечение двигателя постоянного тока (DC motor), иногда называемого двигателем постоянного тока с постоянным магнитом (permanent magnet DC motor, PMDC motor). *Двигатель постоянного тока* состоит из статора (постоянного магнита) и ротора с намотанными на него обмотками. Соединение обмоток ротора осуществляется с помощью щеток, образующих контакт с коммутатором, находящимся на оси, но изолированном от нее. Когда прикладывается постоянное напряжение, ротор вращается, компенсируя магнитное поле статора. Когда поле оказывается скомпенсированным, щетки размыкают контакт с одной обмоткой и образуют контакт с другой. Это меняет полярность магнитного поля ротора на противоположную. Ротор продолжает вращение, пытаясь снова скомпенсировать внешнее магнитное поле статора. Таким образом, ротор вращается, поскольку поле в обмотках переключается согласованно с вращением оси ротора.

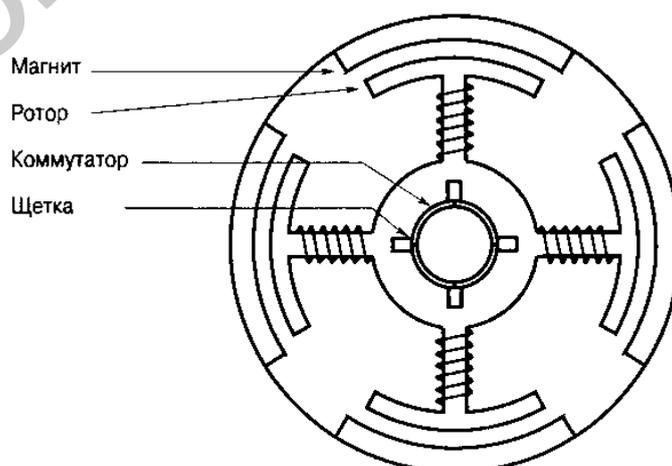


Рисунок 3.1 – Сечение двигателя постоянного тока

На рисунке 3.1 в качестве примера показан ротор с четырьмя секциями, четырьмя щетками и четырьмя коммутирующими контактами. В некоторых двигателях постоянного тока обмотки не наматываются, а выполняются в виде печатных проводников. Это обеспечивает малую инерцию, что позволяет достичь больших ускорений. Кроме того, высокая степень технологичности достигается исключением рутинной проводной намотки обмоток ротора.

Двигатели постоянного тока не теряют синхронизации, как шаговые двигатели. Если нагрузка на ось увеличивается, скорость двигателя снижается вплоть до полной остановки. В микропроцессорных системах двигатели постоянного тока обычно используются совместно с датчиком позиционирования, или энкодером (encoder), который сообщает процессору координаты оси. Энкодеры будут детально рассмотрены в подразделе 3.4.

### 3.2 Управление двигателями постоянного тока

Как и шаговые двигатели, двигатели постоянного тока могут управляться с помощью мостовой схемы прерывания или с помощью аналогового драйвера, выполненного на мощном ОУ. Однако в то время как в управлении шаговым двигателем аналоговый драйвер или прерыватель используется с учетом тока через обмотки, схемотехническое решение контроллера (драйвера) двигателя постоянного тока обычно строится без учета контроля тока в обмотках. Вместо этого контроллер двигателя постоянного тока должен обеспечить достаточный ток для достижения необходимого ускорения (на основе измерений датчика позиционирования – *энкодера*). Если на оси двигателя нагрузка увеличится, то драйвер увеличит силу тока для поддержания заданной скорости. Таким образом, измеряется скорость, а не сила тока. В цепях управления иногда применяется измерение тока моста, однако не для управления, а для контроля над превышением определенной величины тока, соответствующей, например, остановке ротора.

На рисунке 3.2 показан пример работы двигателя постоянного тока при двух различных нагрузках: с легкой нагрузкой – линия 1, с массивной – линия 2. Сначала двигатель разгоняется до постоянной скорости, работает некоторое время, затем замедляется до полной остановки. Как бы то ни было, контроллер снабжает двигатель достаточным током для обеспечения требуемой характеристики скорость/время независимо от нагрузки. По этой причине для приложений с широким диапазоном применяемых нагрузок лучше подходят двигатели постоянного тока.

Одна из интересных особенностей двигателей постоянного тока – возможность управления торможением. Если вручную вращать вал двигателя, получается маленький генератор. Если теперь замкнуть между собой выводы образованного генератора, вращение вала будет затруднено из-за воздействия электромагнитных сил. Таким образом, если замкнуть выводы двигателя во время его вращения, это быстро приведет к его остановке. Такой метод называется *динамическое торможение* (dynamic braking).

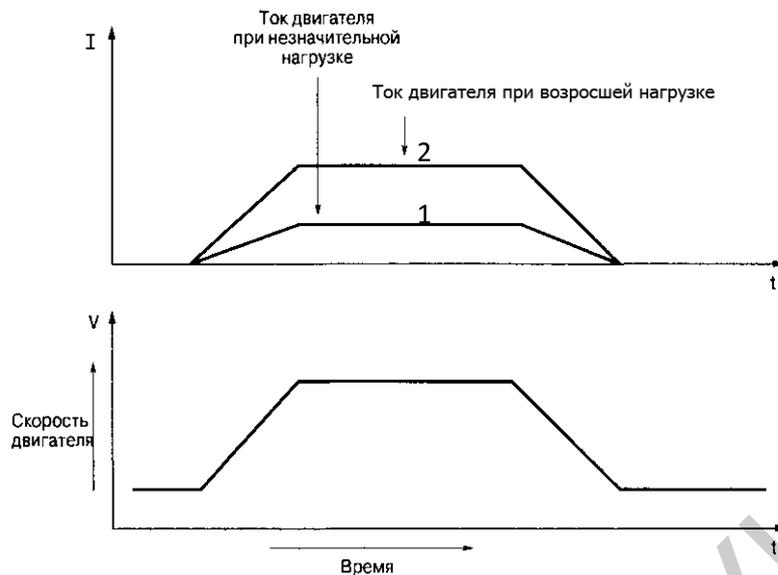


Рисунок 3.2 – Работа двигателя постоянного тока с различными нагрузками

На рисунке 3.3 показана модифицированная мостовая схема. Здесь разделены входы управления двигателем так, что теперь каждый транзистор можно включать по отдельности. Если одновременно установить **ВЫСОКИЙ** уровень на входах А и D, транзисторы VT1 и VT3 откроются, и двигатель будет вращаться в определенном направлении. Если **ВЫСОКИЙ** уровень теперь подать на другую пару транзисторов VT2 и VT4, то двигатель станет вращаться в противоположную сторону.

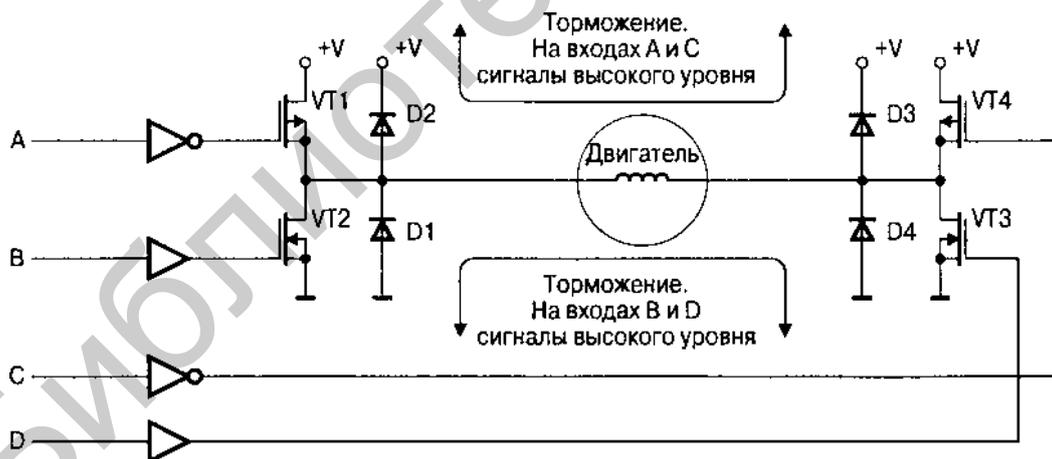


Рисунок 3.3 – Торможение двигателя постоянного тока

Теперь предположим, что двигатель по-прежнему вращается, и на входы В и D подан **НИЗКИЙ** логический уровень, а на входы А и С – **ВЫСОКИЙ**. Тогда откроются транзисторы VT1 и VT4. Допустим, одна сторона двигателя под большим положительным потенциалом, чем другая, например, левая сторона на рисунке 3.3. Ток будет протекать по цепи: через транзистор VT4, обмотку двигателя и диод D2. Это быстро остановит двигатель. Если предположить, что

правая сторона под большим положительным потенциалом, ток потечет через VT1 и D3. Если подать НИЗКИЙ логический уровень на входы В и D вместо А и С, получим тот же самый эффект с протеканием тока через VT3/D1 или VT2/D4.

У многих интегральных схем (ИС) мостовых контроллеров есть опция динамического торможения. Это касается и L6201 и LM18200. ИС L6201 снабжена двумя входами для управления двумя разными половинами моста. Если на оба входа подать одинаковый уровень (ВЫСОКИЙ или НИЗКИЙ), драйвер остановит двигатель. ИС LM18200 имеет отдельный вход для торможения двигателя.

Торможение двигателя может быть применено как для резкого торможения, так и для остановки в определенной позиции. Одно ограничение динамического торможения в том, что его сила не может превысить электромагнитной силы, зависящей от прямого падения напряжения на диоде.

### 3.3 Бесколлекторные двигатели постоянного тока

На рисунке 3.4 показано сечение *бесколлекторного двигателя постоянного тока*. Выглядит довольно похоже на шаговый двигатель, и, действительно, принципы работы бесколлекторного и шаговых двигателей во многом совпадают. Статор данного двигателя состоит из трех катушек (А1/А2, В1/В2 и С1/С2). Катушки соединены по трехфазной схеме с общей точкой. Бесколлекторный двигатель обладает повышенным КПД по сравнению с коллекторным двигателем такого же размера. Катушки в бесколлекторном двигателе прикреплены к корпусу, а не к ротору, что позволяет легче отводить тепло от двигателя.

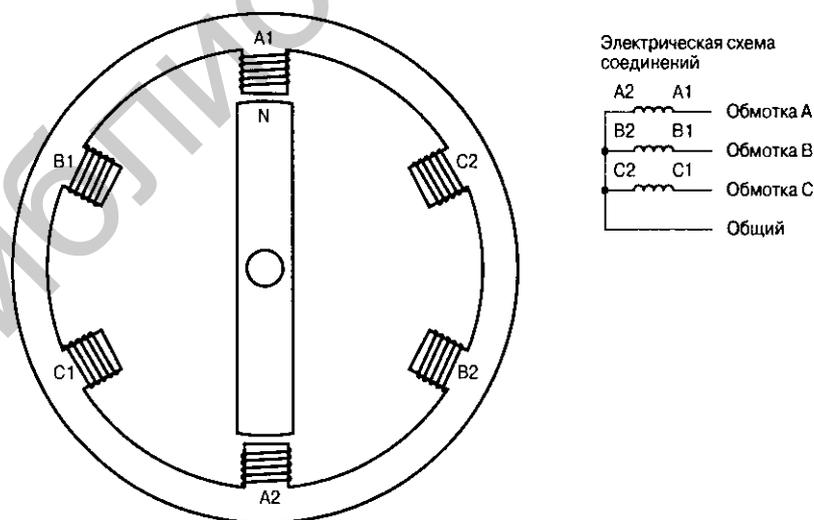


Рисунок 3.4 – Бесколлекторный двигатель постоянного тока

Бесколлекторный двигатель работает точно так же, как двигатель постоянного тока, но без применения щеток. Вместо механической коммутации

для бесколлекторного двигателя требуется внешняя электронная коммутация. Бесколлекторный двигатель может управляться синусоидальным сигналом, но для него более характерно управление дискретной последовательностью постоянных уровней. На рисунке 3.5 показаны обе временные диаграммы. При синусоидальном питании ток можно регулировать, используя режим прерывания или линейную схему. Поскольку обмотки расположены под углом  $120^\circ$  друг относительно друга, синусоидальные сигналы управления должны также иметь сдвиг фаз  $120^\circ$ . Сумма токов трех обмоток равна нулю. Для переключения обычно используются три уровня – высокий, низкий и плавающий (выкл., высокое сопротивление).

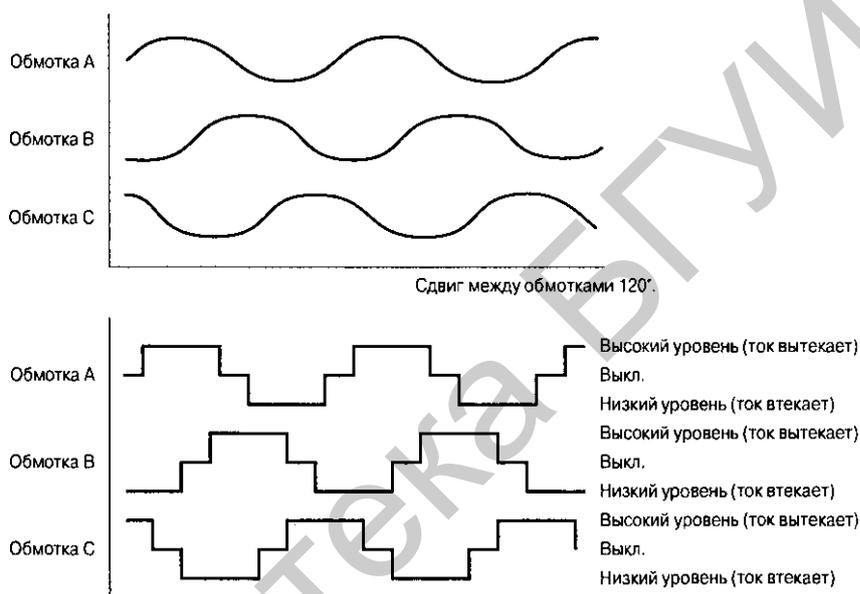


Рисунок 3.5 – Временные диаграммы работы бесколлекторного двигателя

Заметим, что при использовании синусоидального управления, контроллеру не нужен источник питания отрицательной полярности; колебания могут происходить между землей и положительным напряжением питания (или даже между двумя положительными уровнями). Когда все три обмотки находятся под одинаковым потенциалом (например 5 В), через обмотки ток не протекает. Так средняя точка между двумя любыми напряжениями будет образовывать «виртуальную землю» (в данном примере 2,5 В).

Цифровая система управления бесколлекторном двигателем проще, чем шаговым или двигателем со щетками. Поскольку в определенный момент каждая из фаз имеет один только уровень – либо высокий, либо низкий, либо выкл. (высокое сопротивление), для управления не нужна мостовая схема.

На рисунке 3.6 показано как два МОП-транзистора могут быть использованы для управления бесколлекторным двигателем. На входы этой цепи могут быть поданы сигналы либо с контроллера, либо с процессора. Заметим, что защитные диоды нужны и в этой схеме (таблица 3.1).

Таблица 3.1 – Комбинация управляющих сигналов для мостовой схемы

Сигнал		Уровень на обмотке двигателя
управления 1	управления 2	
0	0	Выкл.
0	1	Низкий
1	0	Высокий
1	1	Запрещенное состояние, приведет к замыканию

**Примечание** – Полевой транзистор с каналом  $p$ -типа VT1 открывается, когда на входе 1 – логическая 1, полевой транзистор с каналом  $n$ -типа VT2 открывается, когда на входе 2 – логическая 1.

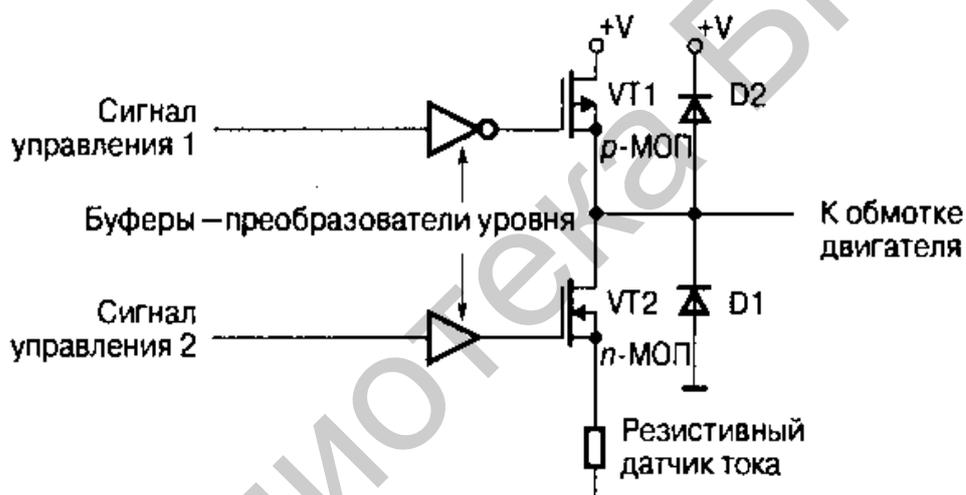


Рисунок 3.6 – Управление бесколлекторным двигателем

Бесколлекторный двигатель обычно оборудован датчиками Холла (от одного до трех) для индикации позиции вала. Однако управлять бесколлекторным двигателем можно и без каких бы то ни было датчиков. Если взглянуть на диаграммы цифровых сигналов, изображенные на рисунке 3.5, можно заметить, что две фазы всегда включены, а одна фаза всегда выключена. Движение ротора будет создавать электродвижущая сила (ЭДС) в обмотке той фазы, которая выключена. Это напряжение пересекает нуль один раз за период и может быть использовано для определения позиции ротора. Заметим, что должно быть измерено напряжение через неиспользуемую обмотку, которая не включена. Другими словами, измеряют разность потенциалов между выводом обмотки и общей точкой соединения всех обмоток.

На рисунке 3.7 показано управление бесколлекторным двигателем без использования датчика. Резисторы  $R1 \dots R6$  образуют делители напряжения для согласования с входным диапазоном аналого-цифрового преобразователя (АЦП). Емкости  $C1 \dots C3$  отфильтровывают помехи сигнала ШИМ-управления. В данной схеме напряжение с общей точки соединения всех трех обмоток используется как напряжение смещения АЦП. Такая схема включения может снизить шумы измерений. Если напряжение на общей точке не удастся использовать как напряжение смещения АЦП, его можно подать на четвертый канал АЦП и из измеряемых значений программно вычитать соответствующую цифровую величину. Если общий провод не выведен за корпус двигателя, значение напряжения в общей точке можно вычислить программно, если позволяет вычислительная мощность процессора. При недостаточной мощности процессора можно требуемые величины вычислить заранее и ввести в постоянное запоминающее устройство (ПЗУ) системы.

При управлении двигателем без использования датчика можно заметить кратковременные выбросы напряжения на измеряемой обмотке в моменты включения и выключения транзисторов. Эти импульсы можно отфильтровать с помощью емкостей, как показано на рисунке 3.7, либо игнорировать измерения в заданные интервалы.

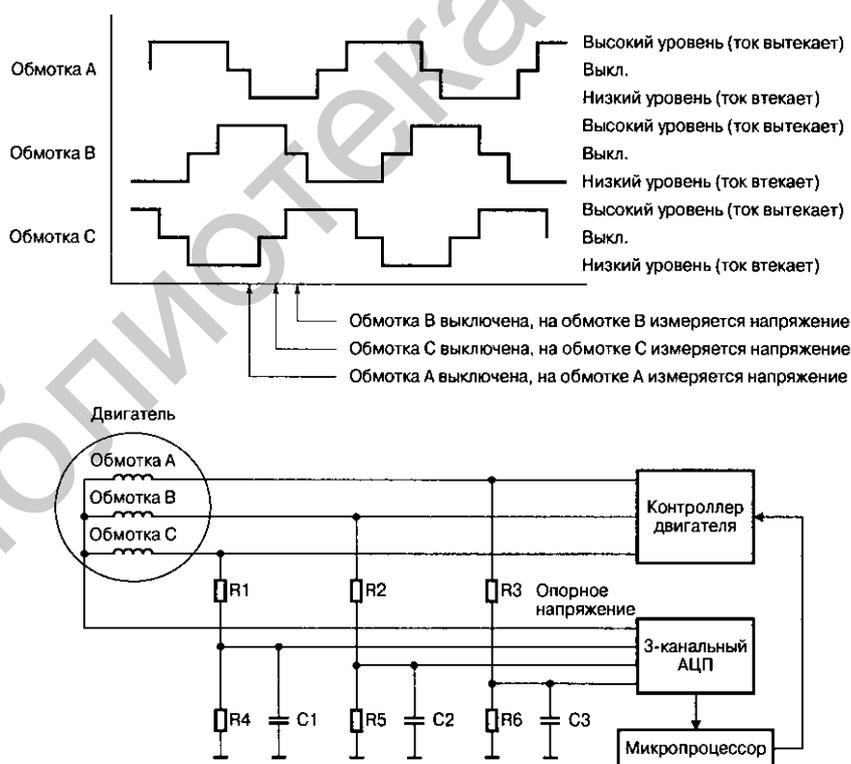


Рисунок 3.7 – Управление бесколлекторным двигателем без использования датчиков

Существует целый ряд *ИС-контроллеров двигателей*, способных управлять *бесколлекторными двигателями* без использования датчиков. К примеру, ИС *TDA5140* фирмы *Philips* может управлять бесколлекторными двигателями с номинальным током в обмотках до 8 А как с использованием датчиков, так и без них.

### 3.4 Энкодеры

Двигатели постоянного тока и бесколлекторные двигатели обычно применяются в микропроцессорных системах совместно со специальными устройствами – *энкодерами* (encoder), укрепляемыми на оси. Назначение этих устройств – передать обратно в систему информацию о позиции оси. Типичный энкодер показан на рисунке 3.8. На данной схеме 4 магнита располагаются вокруг вала двигателя, а на корпус прикрепляется датчик Холла. Датчик Холла будет формировать 4 импульса за один оборот оси.

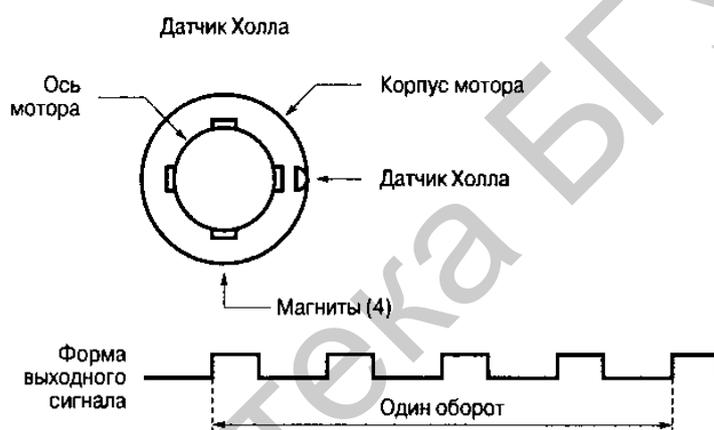


Рисунок 3.8 – Энкодер с датчиком Холла

Четыре импульса за оборот вполне достаточно для управления скоростью вращения в системах с низкой разрешающей способностью, например в вентиляторах. При использовании двигателя с редуктором такой энкодер, установленный на вал двигателя, обеспечит высокое разрешение измерения скорости редуктора. Однако когда требуется измерение скорости вращения двигателя с высокой точностью, используется оптический энкодер.

На рисунке 3.9 показан простой оптический энкодер. На стеклянный диск нанесены непрозрачные метки, в данном примере – 16. Диск устанавливается на ось двигателя, а край его помещается в щелевой оптический датчик. При каждом пересечении датчика меткой фототранзистор закрывается, и оптический датчик формирует импульс. Такой энкодер будет вырабатывать 16 импульсов за один оборот оси. Контроллер может подсчитывать число импульсов для определения угла поворота и числа оборотов.

Этот энкодер, как и энкодер с датчиком Холла, имеет один серьезный недостаток: нельзя определить, в какую сторону вращается двигатель. На рисунке 3.10 показано практическое приспособление для энкодера, дающее

информацию о направлении движения. Снова использован стеклянный диск с непрозрачными метками, но используются уже два оптических датчика, расположенных один возле другого. Длина непрозрачной метки больше, чем расстояние между датчиками. Как только метка попадает в датчик А, на его выходе (канал А) формируется ВЫСОКИЙ логический уровень. И когда метка доходит до датчика В, на его выходе (канал В) также формируется ВЫСОКИЙ логический уровень. При дальнейшем вращении оси сигнал становится НИЗКИМ в канале А, а затем и в канале В.

Если двигатель меняет направление вращения, то первым переключается канал В, а за ним – канал А. Такой *двухканальный энкодер*, называемый также *квадратурным энкодером* (quadrature encoder), предоставляет полную информацию о позиции, скорости и направлении. Типичные энкодеры такого типа способны формировать от 50 до 1000 импульсов на один оборот оси.

Выпускаются также *энкодеры с индексным выходом* (index output). В них используется третий датчик и дополнительная метка, расположенная ближе к центру диска. Метка индекса (см. рисунок 3.10) всего одна на диске и датчик индекса производит один импульс за оборот. Эта опция предоставляет системе информацию об абсолютной стартовой позиции оси двигателя.

На рисунке 3.11 показан трафарет диска абсолютного энкодера (absolute encoder). *Абсолютный энкодер* кодирует непрозрачные метки в двоичные числа, что всегда позволяет судить об абсолютной позиции. Естественно, что в таком энкодере требуется столько же датчиков и колец с метками, каково разрешение датчика. На рисунке показаны четыре внешних кольца. Энкодер с шестью кольцами, например, потребует 6 датчиков и обеспечит разрешение в 64 уникальных кода. Энкодеры абсолютной позиции чрезвычайно дороги. В основном они используются в системах, где необходимо знать положение оси двигателя при включении.

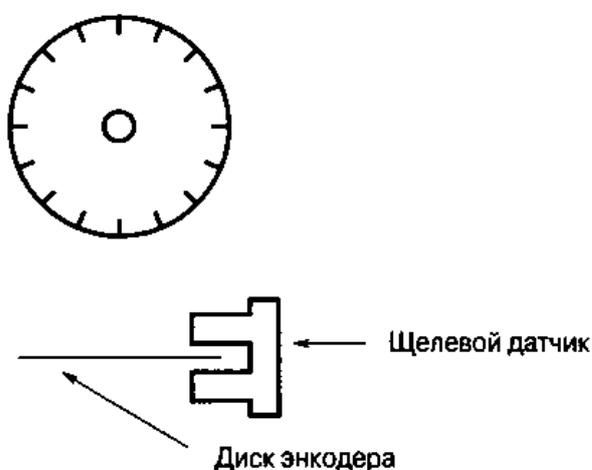


Рисунок 3.9 – Стеклянный диск простого энкодера двигателя

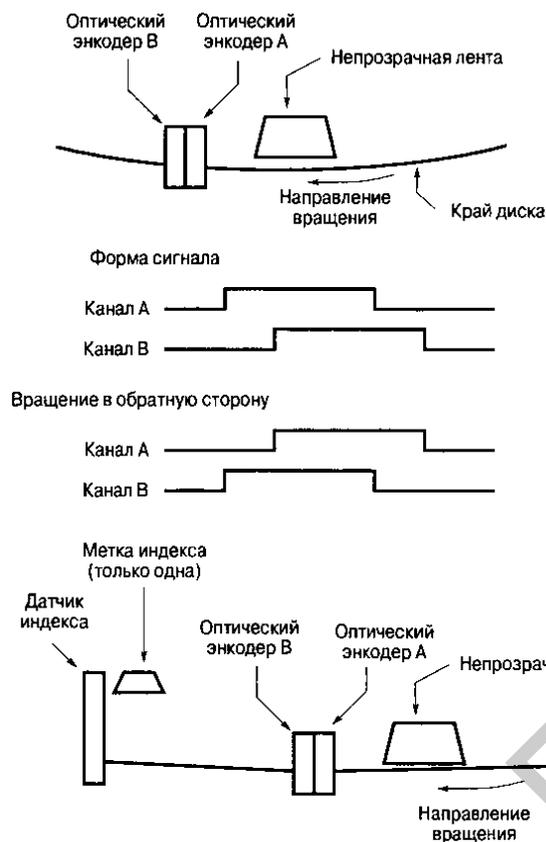


Рисунок 3.10 – Квадратурный энкодер

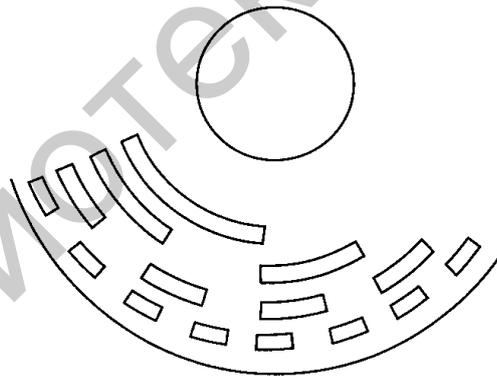


Рисунок 3.11 – Энкодер абсолютной позиции

### 3.5 Управление двигателем

До сих пор действие систем автоматического управления объяснялось на примере нагревателя, что было обусловлено простотой его работы. Описанные выше методы можно также применить и к двигателям, но с некоторыми осложнениями.

На рисунке 3.12 показана ПИД-система управления двигателем. Один из входов системы – цифровое слово, устанавливающее заданную позицию. Позиция двигателя – аналоговая величина (угол поворота оси и т. п.), преобразованная в цифровую форму (содержимое счетчика после подсчета числа импульсов энкодера в результате поворота оси двигателя). Когда ось

двигателя вращается в одном направлении, счетчик считает на сложение, а когда в другом – на вычитание.

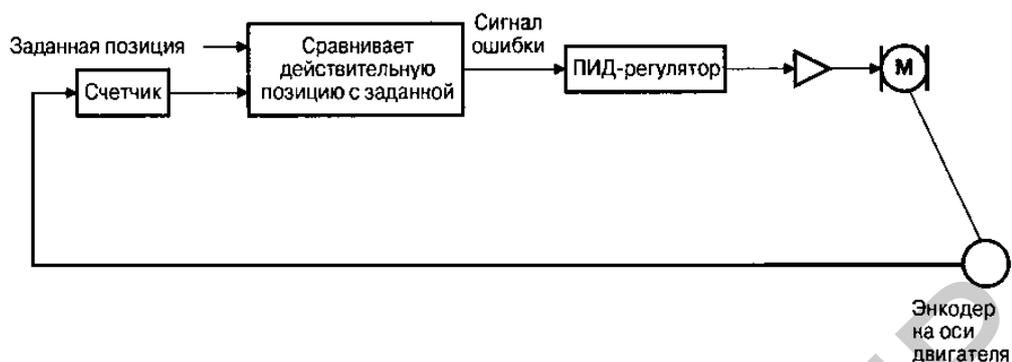


Рисунок 3.12 – ПИД-управление двигателем

Цифровой выход счетчика сравнивается с заданным числом. Полученная разница и является сигналом ошибки (рассогласования). Эта ошибка управления такая же, как в аналоговой системе, за исключением того, что выдается в виде цифрового кода. ПИД-система управления использует эту ошибку (и предыдущие ее значения) для вычисления новой величины.

На рисунке 3.13 показана круговая карусель с 8 дискретными позициями.

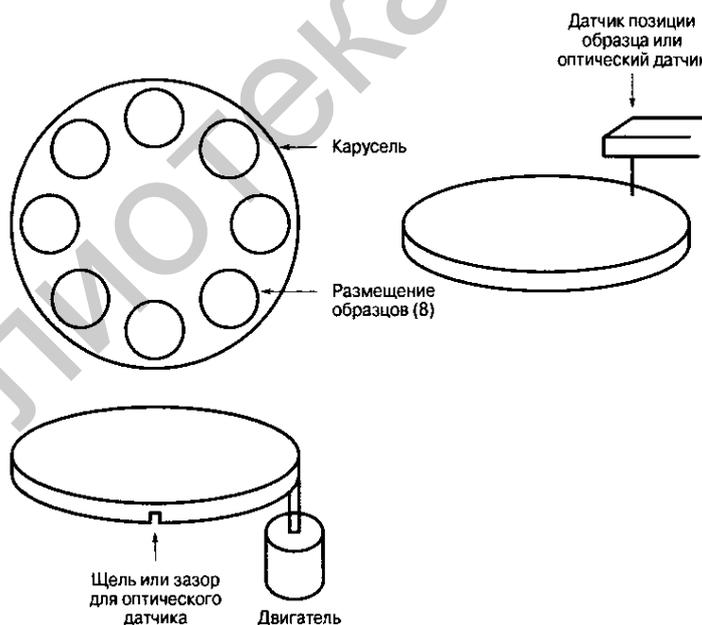


Рисунок 3.13 – Вращающаяся карусель

Такая карусель может использоваться для отправки нужного образца в рабочую зону «руки» манипулятора при химических или медицинских исследованиях либо вращаться непрерывно, например, в системе обработки изображений. Карусель приводится в движение понижающим редуктором, соосным с валом двигателя (на рисунке не показан), снижающим число

оборотов оси двигателя. На один оборот карусели приходится 12 оборотов оси двигателя. Прорезь в карусели и оптопара сообщают системе, когда карусель устанавливается в исходную позицию. Ось двигателя также имеет оптический датчик и энкодер для управления микропроцессором положения оси двигателя.

### 3.5.1 Пример постоянной скорости вращения круговой карусели

Рассмотрим простейший случай применения круговой карусели – для непрерывного вращения. Карусель вращается с постоянной скоростью, которая может быть синхронизирована с оптической системой или видеокамерой. Система управления (пропорциональная или ПИД) задает необходимую скорость вращения оси двигателя. Оптопара необходима для контроля скорости вращения карусели.

Система управления плавно поднимает скорость вращения оси двигателя и далее удерживает ее на определенной величине (рисунок 3.14) до поступления команды «Стоп». Скажем, на один оборот карусели приходится 100 оборотов оси двигателя, и на оси двигателя установлен прецизионный энкодер на 500, что обозначает 500 отсчетов энкодера на один оборот оси двигателя. На рисунке 3.14 период повторения импульсов после делителя показан условно – так, будто приведен каждый сотый импульс.

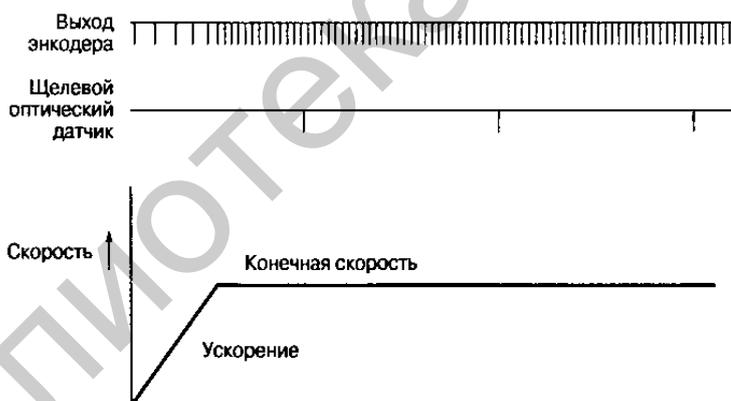


Рисунок 3.14 – Регулировка скорости вращения оси двигателя

Управляющее ПО (контроллера или специализированной микросхемы) обычно проверяет скорость в установившемся режиме (число импульсов встроенного тактового генератора). Не существует никаких указаний при проектировании, чтобы поднять частоту выше частоты генератора ШИМ (если ШИМ используется). На практике такая система приведет к нестабильности, поскольку новый рабочий цикл ШИМ не совпадает с предыдущим.

Вместо деления на равные интервалы можно считывать данные с независимого счетчика при каждом импульсе энкодера. Время между импульсами измеряется, и если скорость низкая, выходной сигнал увеличивается.

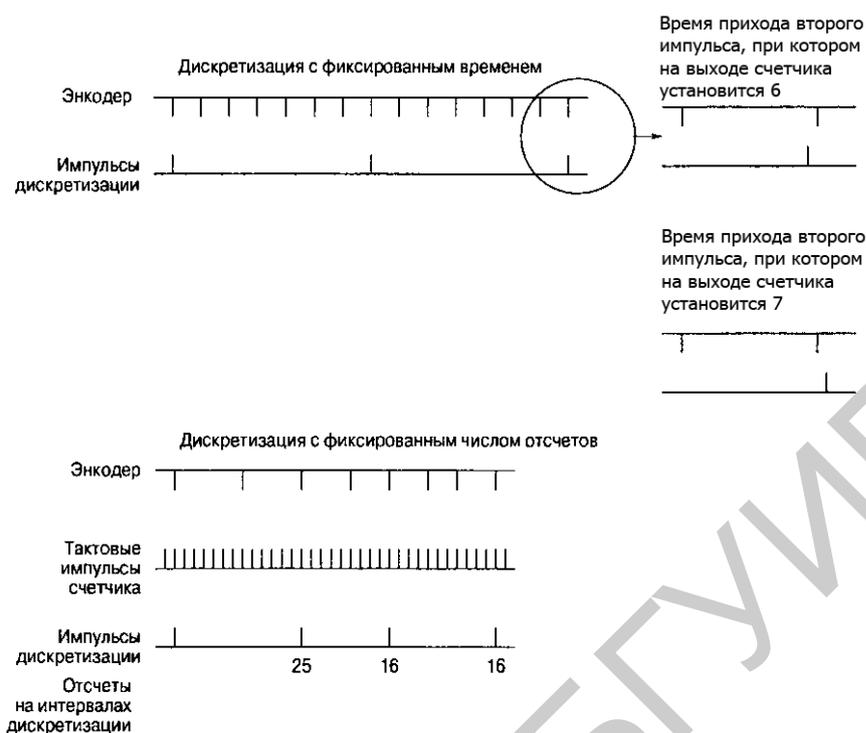


Рисунок 3.15 – Дискретизация скорости вращения оси двигателя

На рисунке 3.15 показана простая диаграмма для обоих методов измерения. При дискретизации с фиксированным временем все сигналы синхронизированы с частотой дискретизации, которая обычно кратна частоте ШИМ. Потенциальный недостаток показан в выделенной кружком области: если импульс кодировщика появится несколько раньше тактового, результат будет отличаться на 1 от случая, когда импульсы придут одновременно. Скорость вращения оси двигателя в обоих случаях почти одинаковая, однако система видит разницу в 1 отсчет. Метод с фиксированным числом отсчетов, дискретизирующий после нескольких отсчетов (3 отсчета на рисунке 3.15), позволяет избежать данной проблемы и увеличить точность управления. Первопричина использования такой системы измерения в том, что частота счетчика гораздо выше, и может потребоваться хранить больше битов данных при медленном вращении оси двигателя. К тому же использование дискретизации с фиксированными отсчетами подразумевает, что интервал дискретизации связан с частотой синхронизации ШИМ – частота дискретизации варьируется с изменением скорости двигателя. Поэтому дискретизация с фиксированным временем применяется чаще.

При применении дискретизации с фиксированным временем необходимо учесть время простоя за счет торможения оси. Если двигатель тормозит, то энкодер не выдает сигналы, и процесса дискретизации не будет. Какой бы метод дискретизации не использовался, карусель в нашем примере будет вращаться с постоянной скоростью с небольшими флуктуациями, зависящими от типа и параметров применяемой системы управления. Поскольку карусель

возвращается в начальное положение после каждого полного оборота, то есть 100 оборотов оси двигателя, индикация начальной позиции будет на каждых 50 000 отсчетах (500 отсчетов энкодера×100 оборотов двигателя/оборот карусели). Так, если первый импульс будет зафиксирован на 10 000-м отсчете, следующий импульс – на отсчете 60 000-м ( $\pm 1$ ). Поэтому, чтобы проверить, следует ли карусель за двигателем, программа могла бы открыть «окно», чтобы зафиксировать импульс около, например, отсчетов 60 000 или 110 000 и т. п.

К тому же счетчики, следящие за позицией карусели, будут периодически переполняться, и ПО (либо специализированная микросхема управления) должно это обстоятельство учитывать.

### 3.5.2 Позиционирование

Рассмотрим более сложный случай дискретного управления нашей каруселью, когда карусель не движется с постоянной скоростью, а устанавливает образец в определенную позицию. После обработки образца позиция карусели должна снова измениться. Типичные осциллограммы для такого типа движения показаны на рисунке 3.16. Двигатель медленно разгоняется до определенной скорости, затем замедляется и останавливается в нужной позиции (с корректным отсчетом энкодера). Трудность здесь в том, чтобы корректно установить все временные соотношения. Системы ПИД-управления движением, используемые в позиционировании, обычно оснащены двумя петлями обратной связи (ОС), работающими параллельно. ПИД-управление устанавливает ток двигателя для достижения необходимой скорости. Входной величиной системы с первой петлей ОС является значение скорости. Система со второй петлей ОС устанавливает форму выходного сигнала в виде трапеции.

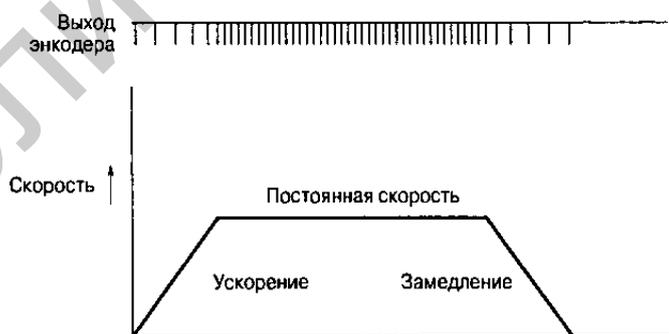


Рисунок 3.16 – Изменение управляющего сигнала в форме трапеции

На рисунке 3.17 показана структурная схема работы системы позиционирования. Микропроцессор посылает команду типа «Перейти на позицию X с ускорением Y». Контур управления позиционированием образца (КУПО) передает информацию о требуемой позиции ПИД-контроллеру. ПИД-контроллер управляет положением оси, регулируя ток двигателя; использует

импульсы энкодера для вычисления текущей позиции. ШИМ-драйвер управляет током двигателя с помощью регулирования рабочего цикла ШИМ. Такой характер работы типичен для ИС, управляющих системами позиционирования, например LM628/629. На этом рисунке блок регулирования скорости двигателя изображен отдельно от микропроцессора, как это обычно реализовано в ИС управления скоростью вращения двигателя. Если бы пришлось писать ПО для микропроцессора или блока цифровой обработки сигналов (ЦОС) прямого управления двигателем, генератор скорости и система ПИД-управления были бы программными функциями.

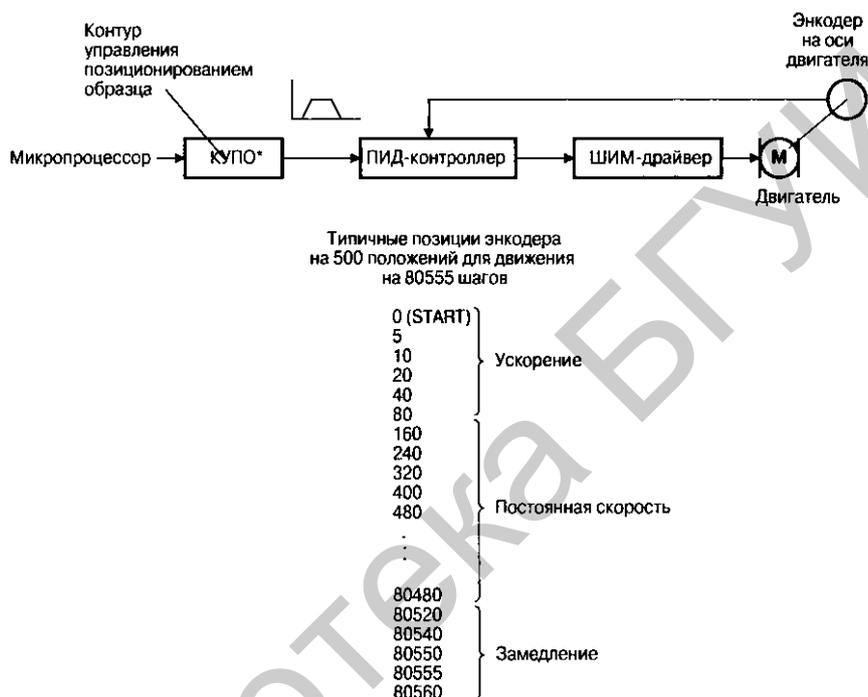


Рисунок 3.17 – Структурная схема системы позиционирования двигателя

Петля ОС управления позицией подает команды позиционирования на систему ПИД-управления. Это та же команда, что была изображена на рисунке 3.14. На рисунке 3.17 показана упрощенная таблица величин позиционирования для движения на 80 555 шагов, то есть немного больше 161 оборота оси двигателя с энкодером на 500 положений. Заметим, что позиция сначала увеличивается на 5 шагов за интервал дискретизации, затем на 10, 20 и т.д. Когда двигатель тормозит, происходит процесс, обратный ускорению, – замедление.

### 3.5.3 Программное обеспечение

Система ПИД-управления устанавливает позицию оси двигателя. Во многих системах нужно обеспечить плавное изменение управляющего сигнала. Необходимо помнить, что данная система механическая и чрезмерное усиление может привести к разрушению ее составных частей.

Наличие постоянного смещения в системе ПИД-управления, где конечная позиция только немного отличается от заданной, может привести к уникальным явлениям в управлении двигателем. В цифровых системах обычно возникают проблемы с неопределенностью в 1 отсчет. Если конечная позиция оси двигателя отличается от заданной на 1 или 2 отсчета и если интегральная составляющая ПИД-системы слишком мала, система может установить чрезмерно большой ток через двигатель и вывести его из строя. Это может произойти потому, что пропорциональная часть системы пытается слегка подтолкнуть ось двигателя до установления окончательного значения, но не обладает достаточным током для этого. Вместо того чтобы выключить питание в конце движения, система устанавливает на двигателе определенный ток. Это может привести к реальным проблемам в системах с большой инерцией или со стопором. Тормозящее действие стопора или большой инерции нужно преодолеть при старте, при этом сила тока двигателя может оказаться довольно высокой. К тому же в двигателе постоянного тока в застопоренном состоянии ток течет только через одну обмотку.

Если в системе нельзя устранить данную проблему только с помощью интегральной составляющей (возможно потому, что не достаточно информации о нагрузке в момент торможения), тогда такое событие должно отслеживать ПО и отключать в определенные моменты времени ток через нагрузку. Если в вашей системе необходимо установить на определенном уровне ток удержания (например, для поддержания манипулятора в вертикальном положении), тогда перепрограммируйте заданную позицию в действительную. Заметим, что генератор позиционирования не знает о реальной позиции двигателя. Предполагается, что система ПИД-управления будет способна обеспечить требуемое ускорение генератора позиционирования. В системе с переменной нагрузкой программе может понадобиться снизить ускорение, если нагрузка слишком большая.

### **3.6 Порядок выполнения работы**

1. Собрать схему, как показано на рисунках 3.18, 3.19.
2. Убедиться, что на панели стенда присутствуют только необходимые для проведения данного опыта переключки, все тумблеры (SA2, SA3, SA5, SA6, SA7, SA8) находятся в нижнем положении («Выключено»). Если необходимо использовать измерительные приборы на панели стенда, должны быть включены тумблеры SA10, SA11.
3. Включить питание стенда (три автоматических выключателя «Сеть»).
4. Подать напряжение на обмотку возбуждения двигателя постоянного тока (ДПТ) (тумблер SA23), плавно вращая регулятор R22 установить номинальный ток возбуждения, ток контролировать по прибору PA2.
5. Включить схему релейно-контакторного управления (тумблер SA70).
6. Подключить исследуемый ДПТ к широтно-импульсному преобразователю (ШИП) (кнопка SB7 4).

7. Установить тип и требуемые настройки регулятора скорости с помощью подстроечных резисторов на панели стенда (R51, R52), а также коэффициент обратной связи  $K_w$ .

8. Установить тип и требуемые настройки регулятора тока с помощью подстроечных резисторов на панели стенда (R61, R62), а также коэффициент обратной связи  $K_i$  (воспользоваться данными настроек из ЛР18).

9. Включить широтно-импульсный преобразователь (тумблер SA22). Плавно изменяя скважность ШИП от 0 до 100 % с помощью регулятора задания (R50), разогнать ДПТ, при этом установить номинальное напряжение, напряжение контролировать по прибору PV1.

10. Включить ШИП нагрузочной машины M2 тумблером SA25.

11. Плавно изменяя ток нагрузочной машины регулятором R6 (контролировать по прибору PA3), снять зависимость скорости и тока исследуемого двигателя от нагрузки. При выполнении опыта ток электродвигателей не должен превышать номинальный. Данные занести в таблицу 3.2.

12. Отключить исследуемый ДПТ от ШИП (кнопка SB7 3).

13. Повторить пункты 7...12 для различных значений величин напряжения на якоре. Данные занести в таблицу 3.2.

14. Перевести регуляторы R6, R21, R22 в крайнее левое положение.

15. Выключить тумблеры SA21, SA22, SA23, SA25, SA70.

16. Выключить питание (три автоматических выключателя «Сеть»).

17. Разобрать схему.

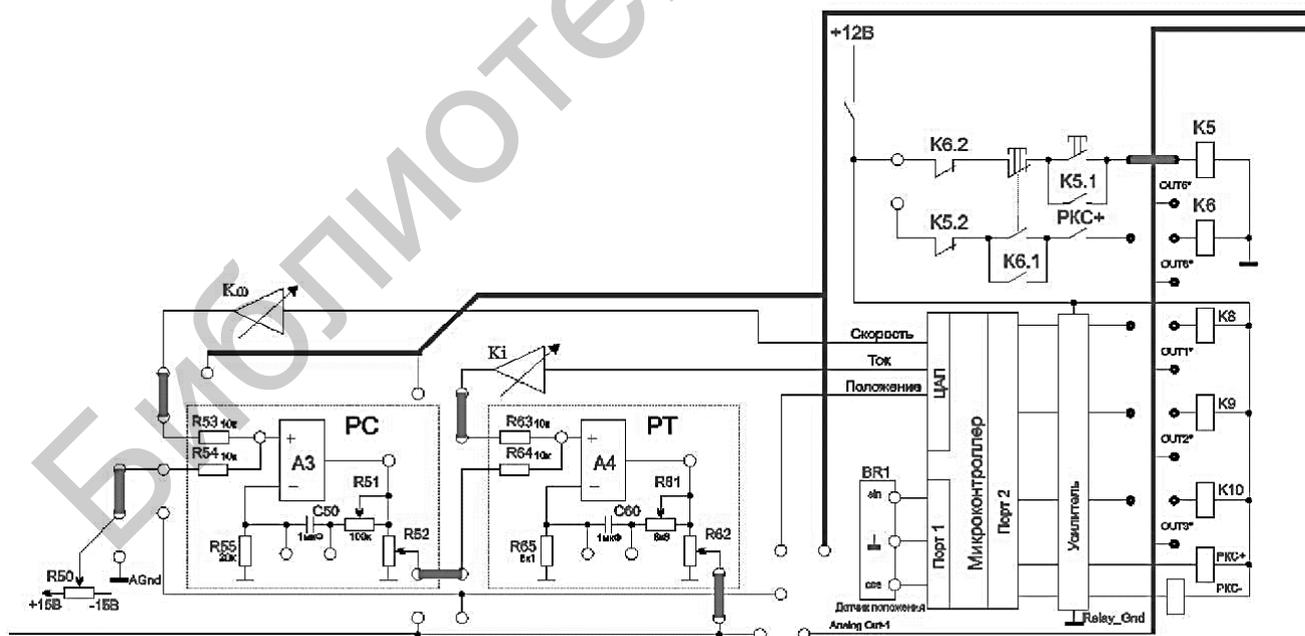


Рисунок 3.18 – Схема системы управления для исследования скоростных и механических характеристик ДПТ в замкнутой САУ

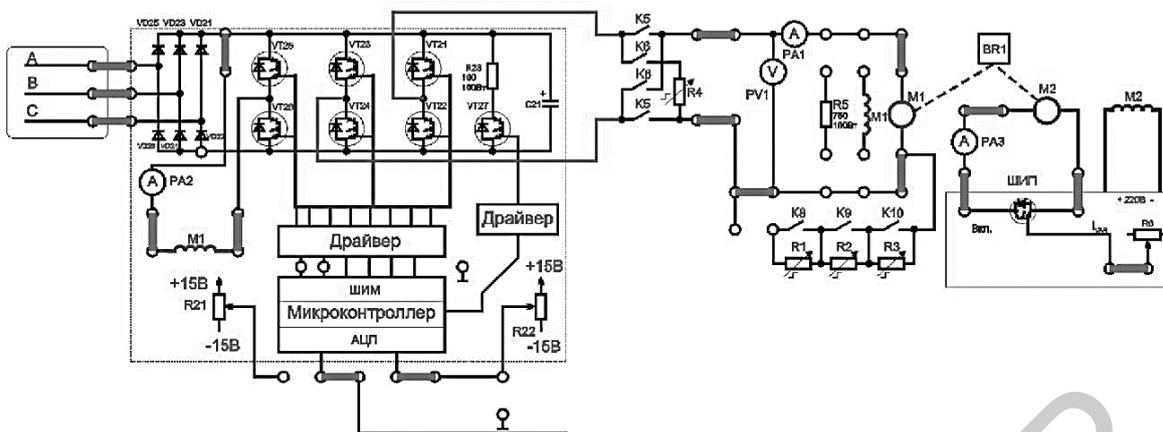


Рисунок 3.19 – Схема силовой части и подключения сигналов задания для исследования статических характеристик ДПТ

Таблица 3.2 – Результаты измерения параметров ДПТ

Установленное напряжение якоря $U = \underline{\hspace{2cm}}$ В					
№	Измерено				Вычислено
	$I_{\text{нагр}}, \text{А}$	$\omega, \text{рад/с}$	$U, \text{В}$	$I, \text{А}$	$M_{\text{нагр}}, \text{Н}\cdot\text{м}$
1					
2					
3					
4					
5					
...					

### 3.7 Контрольные вопросы

1. Как устроен двигатель постоянного тока?
2. Каковы особенности управления двигателями постоянного тока?
3. Каковы отличия управляющих сигналов для разгона и торможения ДПТ?
4. Как устроен бесколлекторный двигатель постоянного тока?
5. Для чего применяются энкодеры при управлении ДПТ?
6. Как устроен энкодер?
7. Каковы особенности программной реализации управления ДПТ?