


Архитектура универсальных программируемых ЛОГИЧЕСКИХ КОНТРОЛЛЕРОВ



Что это такое «промышленный контроллер»?

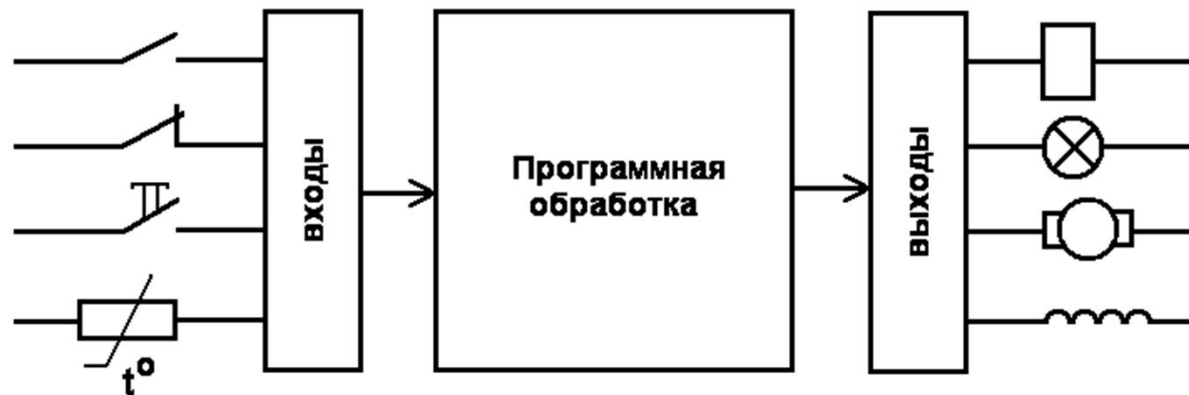
Термином "**промышленный контроллер**" обозначают микропроцессорное устройство со встроенным аппаратным и программным обеспечением, которое используется для выполнения функций управления технологическим оборудованием.

Специализированный контроллер это устройство, которое разрабатывалось для конкретного применения и не может применяться в иных местах.

Универсальный контроллер строится исходя из концепции размещения в рамках выбранного конструктива максимально возможного и избыточного для отдельного конкретного применения набора аппаратных средств или из обеспечения конструктивной возможности набора (компоновки) заданного набора средств. Современный универсальный контроллер при этом чаще всего состоит из базового блока, включающего в себя процессорный модуль (CPU), и набора периферийных модулей, состав которых будет обеспечивать выполнение заданного набора функций. Другим фактором универсализации всегда выступает программное управление.

Обобщенная структура контроллера

ПЛК представляют собой микропроцессорное устройство, предназначенное для сбора, преобразования, обработки, хранения информации и выработки команд управления. Каждый из них представляет собой вычислительное устройство, имеющее некоторое множество входов и выходов. Контроллер отслеживает изменение входов и вырабатывает программно определяемое воздействие на выходах.



Контроллер имеет:

- дискретные входы и выходы
- аналоговые входы и выходы
- специальные входы и выходы
- системное и прикладное программное обеспечение

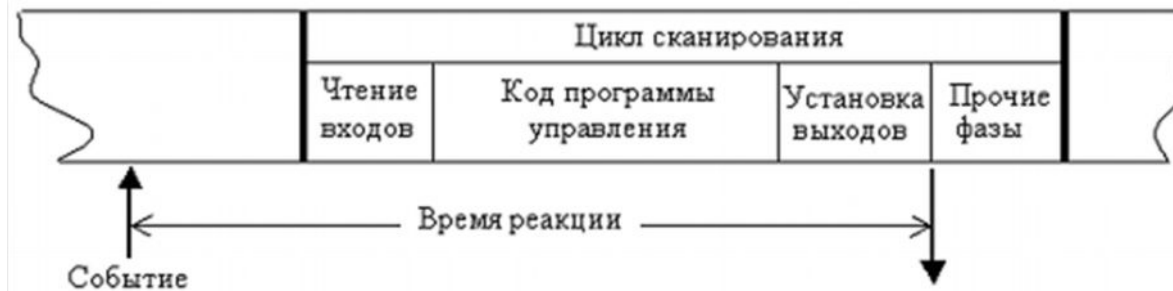
Системное и прикладное программное обеспечение. Рабочий цикл ПЛК.

Программное обеспечение универсальных контроллеров состоит из двух частей. Первая часть это системное программное обеспечение. Проводя аналогию с программным обеспечением ЭВМ можно сказать, что оно выполняет функции операционной системы, т.е. управляет работой узлов контроллера, занимается организацией их взаимосвязи, внутренней диагностикой. Это программное обеспечение расположено в постоянной памяти в адресном пространстве центрального процессора и всегда готово к работе.

Другая часть программного обеспечения универсальных контроллеров это прикладные программы управления конкретным процессом. Эти программы создаются разработчиком системы управления.

ПЛК работают циклически. Рабочий цикл ПЛК включает 4 фазы:

- опрос входов;
- выполнение пользовательской программы;
- установку значений выходов;
- вспомогательные операции (диагностика, подготовка данных для отладчика, визуализации и т. д.).





Функциональные ресурсы контроллеров

При работе с контроллером пользователю предоставляются различные функциональные ресурсы. Ресурсы используются для управления процессом и определения его текущего состояния.

В большинстве контроллеров имеются следующие ресурсы:

- цифровые входы (I);
- цифровые выходы (Q);
- аналоговые входы (AIW);
- аналоговые выходы (AQW);
- память переменных (V);
- меркеры и специальные меркеры (M, SM);
- таймеры (T);
- счетчики (C);
- быстрые счетчики (HC);
- аккумуляторы (AC);
- аналоговые потенциометры;
- часы реального времени.

В программах можно использовать метки переходов, подпрограммы, подпрограммы обработки прерываний

Для обращения в программе к ресурсам всем им присвоены идентификаторы (обозначения). Состояние ресурсов (соответствующих переменных) отображается в специально выделенных областях памяти. Например, если напряжение, приложенное к входу равно “1”, то этот сигнал отображается в ячейку памяти, связанную с этим входом. При обращении к ресурсам можно пользоваться прямой и косвенной адресацией. С помощью прямой адресации памяти, адреса можно указывать в явном виде. Благодаря этому программа имеет прямой доступ к информации. Адрес состоит из идентификатора области памяти, адреса байта, а также номера бита (такая адресация называется также адресацией “байт.бит”). Если необходимо обратиться к адресу косвенно, то вначале создают указатель, указывающий на этот адрес. Указатели являются двойными словами. Для создания указателя используется операция специальные команды.

Обращение к входам-выходам. Отображение входов-выходов

Цифровые входы I0.0 – I... В начале каждого цикла CPU опрашивает физические входы и записывает эти значения в область отображения процесса на входах (в память). Можно обращаться к этой области отображения процесса в формате бита, байта, слова или двойного слова.

Цифровые выходы Q0.0 - Q... В конце цикла CPU копирует значения из области отображения процесса на выходах на физические выходы. Можно обращаться к этой области отображения процесса в формате бита, байта, слова или двойного слова.

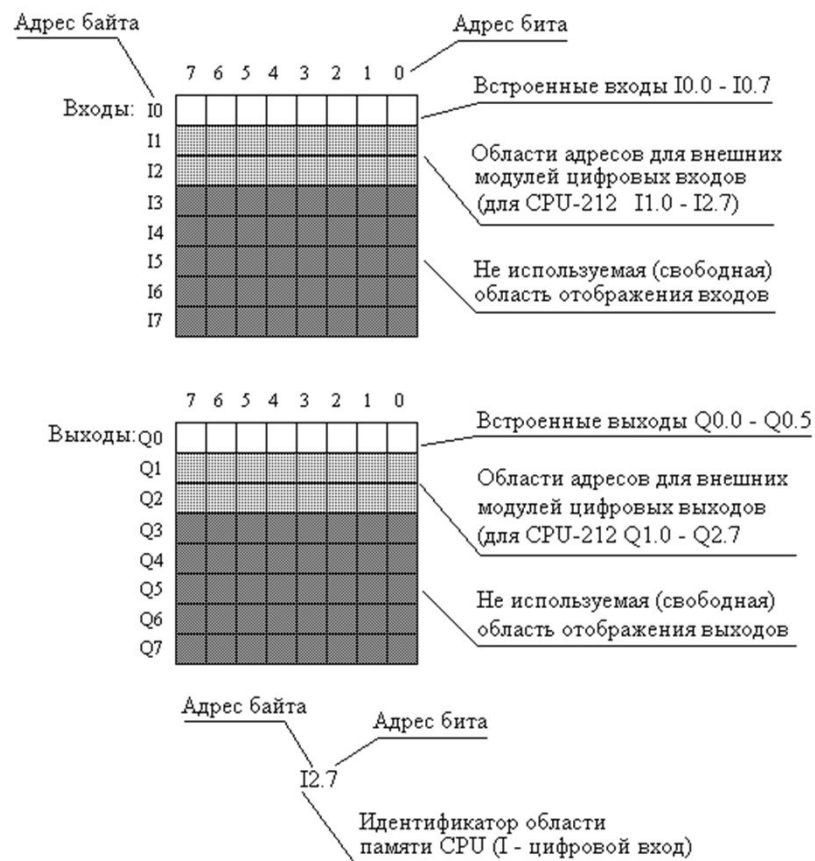
Аналоговые входы – AIW0 - AIW... CPU преобразует аналоговые значения в цифровые слова с разрядностью слова (16 битов).

Аналоговые выходы – AQW0 - AQW... CPU преобразует цифровые значения с разрядностью слова (16 битов) в ток или напряжение пропорционально цифровому значению.

Области памяти, формируемые по внешним сигналам (цифровые и аналоговые входы и выходы) называются областью отображения входов и областью отображения выходов. Эти области памяти содержат образ сигналов от процесса.

Встроенные входы и выходы в центральном устройстве имеют фиксированные адреса.

При необходимости CPU можно дополнить входами и выходами, присоединяя модули расширения. Адреса входов и выходов в модуле расширения определяются видом входов и выходов, а в случае нескольких модулей одинакового типа – также расположением модуля относительно других. В частности, модуль вывода не влияет на адреса входов в модуле ввода и наоборот. Адреса входов и выходов аналоговых и цифровых модулей также не зависят друг от друга.



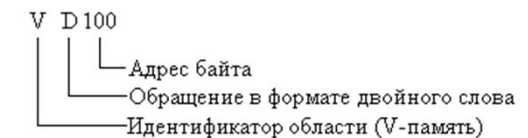
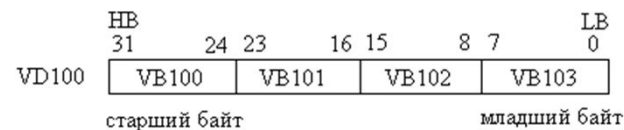
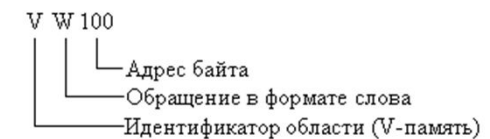
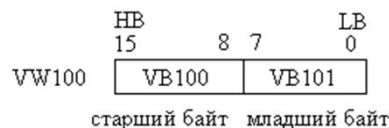
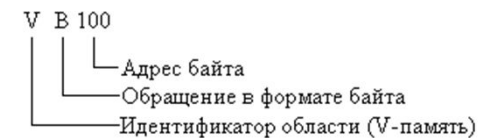
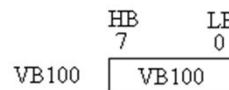
Обращение к переменным. Области переменных.

Память переменных V0.0 - V... (в том числе, область, устойчивая к нулевому напряжению). В памяти переменных можно хранить промежуточные результаты, рассчитываемые операциями программы. К памяти переменных можно обращаться в формате бита, байта, слова или двойного слова.

Меркеры M0.0 - M... (в том числе, область, устойчивая к нулевому напряжению). Внутренние меркеры можно использовать как управляющие реле для того, чтобы сохранять промежуточные результаты операций или другую управляющую информацию. Можно обращаться к меркерам в формате бита, байта, слова или двойного слова.

Специальные меркеры SM0.0 - SM... (в том числе защищенные от записи). С помощью специальных меркеров можно производить обмен информацией между CPU и программой. Кроме того, специальные меркеры служат для того, чтобы выбирать особые функции CPU и управлять ими.

Данные в различных областях памяти представляются как байт, слово, двойное слово. Старший разряд в слове используется для обозначения знака. Байт имеет байтовый адрес и битовые адреса с 0 по 7. Группа из двух байт называется словом. Если необходимо обратиться к байту, слову или двойному слову, то указывается идентификатор области, формат доступа и начальный адрес значения в формате байта, слова или двойного слова.





Сложные типы данных и обращение к ним

Таймеры T0 - T... являются элементами, подсчитывающими приращения времени. Таймеры имеют разрешающую способность (приращения, определяемые базой времени) 1 мс, 10мс и 100 мс.

Каждый таймер имеет в своем распоряжении следующие две переменные: текущее значение: это целое число (16 битов) со знаком, хранит значение времени таймера; бит таймера: этот бит включается (устанавливается в "1"), когда текущее значение таймера больше или равно предварительно установленному значению.

Счетчики C0 - C... В CPU счетчики являются элементами, подсчитывающими нарастающие фронты сигнала на счетных входах. CPU имеет в своем распоряжении счетчики двух видов: счетчик первого вида считает только вперед, тогда как счетчик другого вида считает как вперед, так и назад. Реализуются на программном уровне, как элементы системного программного обеспечения.

Каждый счетчик имеет в своем распоряжении следующие две переменные: текущее значение: это целое число (16 битов) со знаком хранит накопленное значение счетчика; бит счетчика: этот бит включается (устанавливается в "1"), когда текущее значение счетчика больше или равно предварительно установленному значению.

Быстрые счетчики HC0 - HC... Быстрые счетчики подсчитывают события быстрее, чем CPU может опрашивать эти события.

Быстрые счетчики имеют в своем распоряжении 32-битное счетное значение (текущее значение). Программа управляет направлением счета (вперед или назад) через бит управления направлением. Максимальная частота счета быстрых счетчиков составляет несколько килogerц.

Обращение к сложным типам данных производится указанием в адресе идентификатора области и номера элемента.



Аккумуляторы и аналоговые потенциометры

Аккумуляторы АС0 - АС... Аккумуляторы являются элементами для чтения/записи, которые используются как память. С помощью аккумуляторов можно, например, передавать параметры в подпрограмму, а также принимать их обратно, или сохранять промежуточные результаты вычислений.

Аналоговые потенциометры. СPU может иметь в своем распоряжении один или два аналоговых потенциометра (под откидной крышкой СPU). С помощью этих потенциометров можно увеличивать и уменьшать значения, записанные в специальные ячейки (меркеры). Эти защищенные от записи значения могут служить в программе для ряда функций, например, при актуализации текущих значений таймеров и счетчиков, при вводе или изменении предварительно установленных значений или при установке граничных значений. Аналоговые потенциометры имеют в своем распоряжении номинальный диапазон (обычно от 0 до 255). Аналоговые потенциометры можно настраивать вручную на передней панели СPU с помощью маленькой отвертки: вращая вправо, если необходимо увеличить значение аналогового потенциометра, или влево для того, чтобы уменьшить значение.



Поддержка прерываний

Контроллеры поддерживает прерывания от ввода/вывода, коммуникационных портов, прерывания, управляемые временем. К прерываниям от ввода/вывода относятся прерывания при нарастающем или спадающем фронте, прерывания от быстрых счетчиков и прерывания от последовательности импульсов. CPU может создавать прерывание при нарастающем и/или спадающем фронте на цифровом входе. Прерывания передачи и приема облегчают коммуникацию с программным управлением. С помощью прерывания, управляемого временем, можно определять действия, которые должны выполняться периодически. Управляемое временем событие прерывания вызывает соответствующую программу обработки прерываний каждый раз, когда истекает время. С помощью управляемого временем события прерывания управляют регулярным опросом аналоговых входов.

Прерывания обрабатываются контроллером в пределах соответствующих им приоритетов в последовательности их появления. Приоритеты прерываний назначаются согласно следующей фиксированной схеме приоритетов:

- коммуникационные прерывания – высший приоритет;
- прерывания от ввода/вывода (включая быстрые счетчики);
- управляемые временем прерывания – низший приоритет.

Если в программе используются прерывания, то программы обработки прерываний, поставленные в соответствие событиям прерываний, запоминаются как часть главной программы. Однако программы обработки прерываний не обрабатываются как составная часть нормального цикла, а обрабатываются только тогда, когда появляется событие прерывания. Это возможно в любой точке цикла, т.е. CPU обрабатывает разблокированные прерывания асинхронно по отношению к циклу. Обработка прерываний происходит в порядке их появления и в соответствии с их приоритетом.

В различных контроллерах реакция на вложенные по времени прерывания может организована по разному.

Например в простейшем, но достаточно распространенном варианте система прерываний организуется следующим образом: всегда активна только одна программа обработки прерываний; если в данный момент обрабатывается программа обработки прерываний, то эта программа доводится до конца; она не может прерываться программой обработки прерываний, появляющейся позже, даже если приоритет этой программы выше; прерывания, возникающие вовремя обработки другого прерывания, принимаются в очередь и обрабатываются позже.



Функциональные ресурсы и области памяти CPU S7-200

Описание	CPU 212	CPU 214	CPU 215	CPU 216
Размер программы пользователя	512 слов	2 К слов	4 К слов	4 К слов
Размер данных пользователя	512 слов	2 К слов	2,5 К слов	2,5 К слов
Отображение процесса на входах	E0.0 - E7.7	E0.0 - E7.7	E0.0 - E7.7	E0.0 - E7.7
Отображение процесса на выходах	A0.0 - A7.7	A0.0 - A7.7	A0.0 - A7.7	A0.0 - A7.7
Аналоговые входы (защищенные от записи)	AEW0 - AEW30	AEW0 - AEW30	AEW0 - AEW30	AEW0 - AEW30
Аналоговые выходы (защищенные от записи)	AAW0 - AAW30	AAW0 - AAW30	AAW0 - AAW30	AAW0 - AAW30
Память переменных (V) Область, устойчивая к нулевому напряжению (макс.)	V0.0 - V1023.7 V0.0 - V199.7	V0.0 - V4095.7 V0.0 - V1023.7	V0.0 - V5119.7 V0.0 - V5119.7	V0.0 - V5119.7 V0.0 - V5119.7
Меркеры (M) Область, устойчивая к нулевому напряжению (макс.)	M0.0 - M15.7 MB0 - MB13	M0.0 - M31.7 MB0 - MB13	M0.0 - M31.7 MB0 - MB13	M0.0 - M31.7 MB0 - MB13
Специальные меркеры (SM) Защищенные от записи	SM0.0 - SM45.7 SM0.0 - SM29.7	SM0.0 - SM85.7 SM0.0 - SM29.7	SM0.0 - SM194.7 SM0.0 - SM29.7	SM0.0 - SM194.7 SM0.0 - SM29.7
Таймеры Формирование задержки включения с запоминанием 1 мс	64 (T0 - T63) T0	128 (T0 - T127) T0, T64	256 (T0 - T255) T0, T64	256 (T0 - T255) T0, T64
Формирование задержки включения с запоминанием 10 мс	T1 - T4	T1 - T4, T65 - T68	T1 - T4, T65 - T68	T1 - T4, T65 - T68
Формирование задержки включения с запоминанием 100 мс	T5 - T31	T5 - T31, T69 - T95	T5 - T31, T69 - T95	T5 - T31, T69 - T95
Формирование задержки включения 1 мс	T32	T32, T96	T32, T96	T32, T96
Формирование задержки включения 10 мс	T33 - T36	T33-T36, T97-T100	T33-T36, T97-T100	T33-T36, T97-T100
Формирование задержки включения 100 мс	T37 - T63	T37-T63, T101-T127	T37-T63, T101-T255	T37-T63, T101-T255
Счетчики	Z0 - Z63	Z0 - C127	Z0 - C255	Z0 - C255
Быстрые счетчики	HC0	HC0 - HC2	HC0 - HC2	HC0 - HC2
Реле шагового управления	S0.0 - S7.7	S0.0 - S15.7	S0.0 - S31.7	S0.0 - S31.7
Аккумуляторы	AC0 - AC3	AC0 - AC3	AC0 - AC3	AC0 - AC3
Переходы/Метки перехода	0 - 63	0 - 255	0 - 255	0 - 255
Вызовы/Подпрограммы	0 - 15	0 - 63	0 - 63	0 - 63
Программы обработки прерываний	0 - 31	0 - 127	0 - 127	0 - 127
События прерываний	0, 1, 8 - 10, 12	0 - 20	0 - 23	0 - 26
PID-регулятор	не поддержив.	не поддерживается	0 - 7	0 - 7
Порты	0	0	0	0 и 1



Области операндов CPU-S7-200

Формат доступа	CPU 212	CPU 214	CPU 215	CPU 216
Бит (Байт.Бит)	V 0.0 - 1023.7 E 0.0 - 7.7 A 0.0 - 7.7 M 0.0 - 15.7 SM 0.0 - 45.7 T 0 - 63 Z 0 - 63 S 0.0 - 7.7	V 0.0 - 4095.7 E 0.0 - 7.7 A 0.0 - 7.7 M 0.0 - 31.7 SM 0.0 - 85.7 T 0 - 127 Z 0 - 127 S 0.0 - 15.7	V 0.0 - 5119.7 E 0.0 - 7.7 A 0.0 - 7.7 M 0.0 - 31.7 SM 0.0 - 194.7 T 0 - 255 Z 0 - 255 S 0.0 - 31.7	V 0.0 - 5119.7 E 0.0 - 7.7 A 0.0 - 7.7 M 0.0 - 31.7 SM 0.0 - 194.7 T 0 - 255 Z 0 - 255 S 0.0 - 31.7
Байт	VB 0 - 1023 EB 0 - 7 AB 0 - 7 MB 0 - 15 SMB 0 - 45 AC 0 - 3 SB 0 - 7 Константа	VB 0 - 4095 EB 0 - 7 AB 0 - 7 MB 0 - 31 SMB 0 - 85 AC 0 - 3 SB 0 - 15 Константа	VB 0 - 5119 EB 0 - 7 AB 0 - 7 MB 0 - 31 SMB 0 - 194 AC 0 - 3 SB 0 - 31 Константа	VB 0 - 5119 EB 0 - 7 AB 0 - 7 MB 0 - 31 SMB 0 - 194 AC 0 - 3 SB 0 - 31 Константа
Слово	VW 0 - 1022 T 0 - 63 Z 0 - 63 EW 0 - 6 AW 0 - 6 MW 0 - 14 SMW 0 - 44 AC 0 - 3 AEW 0 - 30 AAW 0 - 30 SW 0 - 6 Константа	VW 0 - 4094 T 0 - 127 Z 0 - 127 EW 0 - 6 AW 0 - 6 MW 0 - 30 SMW 0 - 84 AC 0 - 3 AEW 0 - 30 AAW 0 - 30 SW 0 - 14 Константа	VW 0 - 5118 T 0 - 255 Z 0 - 255 EW 0 - 6 AW 0 - 6 MW 0 - 30 SMW 0 - 193 AC 0 - 3 AEW 0 - 30 AAW 0 - 30 SW 0 - 30 Константа	VW 0 - 5118 T 0 - 255 Z 0 - 255 EW 0 - 6 AW 0 - 6 MW 0 - 30 SMW 0 - 193 AC 0 - 3 AEW 0 - 30 AAW 0 - 30 SW 0 - 30 Константа
Двойное слово	VD 0 - 1020 ED 0 - 4 AD 0 - 4 MD 0 - 12 SMD 0 - 42 AC 0 - 3 HC 0 SD 0 - 4 Константа	VD 0 - 4092 ED 0 - 4 AD 0 - 4 MD 0 - 28 SMD 0 - 82 AC 0 - 3 HC 0 - 2 SD 0 - 12 Константа	VD 0 - 5116 ED 0 - 4 AD 0 - 4 MD 0 - 28 SMD 0 - 191 AC 0 - 3 HC 0 - 2 SD 0 - 28 Константа	VD 0 - 5116 ED 0 - 4 AD 0 - 4 MD 0 - 28 SMD 0 - 191 AC 0 - 3 HC 0 - 2 SD 0 - 28 Константа