

## Доступ к данным Logix5000

### Назначение

Этот документ описывает доступ к данным контроллера Logix5000, используя следующие методы:

- Сервисы CIP (способ связи, присущий Logix5000);
- Команды связи программируемого контроллера (PCCC), для совместимости с контроллерами PLC и SLC:
  - команды PLC-2,
  - команды PLC-5,
  - команды SLC.

### Когда использовать этот документ

Используйте этот документ, если вам нужно записать данные в контроллер или прочитать данные из контроллера Logix5000, используя внешнее устройство.

### Как использовать этот документ

Этот документ использует следующие сокращения и определения:

Этот термин:	Означает:
PLC-5	тип программируемого логического контроллера Allen-Bradley
PLC-2	устаревший тип программируемого логического контроллера Allen-Bradley
SLC	малый логический контроллер Allen-Bradley
CIP	информационно-управляющий протокол
GEN STS	основной статус сообщения CIP
IOI	внутренний идентификатор объекта (т.е. как адресовать данные в сообщениях CIP)
segment (сегмент)	часть IOI
tag	символьное имя данных
STS / EXT STS	состояние передачи сообщения
TNS	уникальный 16-битовый идентификатор транзакции
address offset (смещение адреса)	2-х байтовое поле, которое указывает смещение байта от начала файла или тега
system address (системный адрес)	многобайтовое поле, которое указывает где начать операцию
packet offset (смещение пакета)	указывает, как долго продолжать операцию за системным адресом
total transaction (транзакция – «всего»)	указывает общую протяженность предполагаемой операции

Для дополнительной информации, обратитесь к «Справочнику по протоколу DF1 и набору команд», публикация 1770-6.5.16.

## Важная информация для пользователя

Из-за различных возможностей использования продуктов, описанных в этой публикации, тот, кто ответственен за приложение и использование этого оборудования управления, должен убедиться сам, что были приняты все необходимые меры чтобы гарантировать, что каждое приложение и использование удовлетворяет всем требованиям безопасности и производительности, включая любые применимые законы, нормы, коды и стандарты.

Иллюстрации, диаграммы, образцы программ и примеры подключения, показанные в этом руководстве, приведены исключительно для примера. Так как существуют многие отличия и требования связанные с каждой конкретной установкой, Allen-Bradley не принимает ответственность (в том числе и ответственность за интеллектуальную собственность) за фактическое использование основанное на примерах, приведенных в этой публикации.

Публикация Allen-Bradley SGI-1.1, «Руководство по безопасности для приложений, установки и эксплуатации полупроводникового оборудования управления» (доступно в вашем локальном офисе Allen-Bradley), описывает некоторые важные различия между полупроводниковым оборудованием и электромеханическими устройствами, которые должно быть приняты во внимание при применении продуктов, например, тех, что описаны в этой публикации.

Воспроизведение содержания этой, охраняемой авторским правом публикации, в целом или частично, без письменного разрешения Rockwell Automation, запрещено.

В этом руководстве мы используем примечания, чтобы уведомить вас о соображениях безопасности:

---

**ВНИМАНИЕ**

Идентифицирует информацию о методах или обстоятельствах, которые могут привести к травме или смерти персонала, ущербу собственности или экономическому убытку.

---

Примечания «внимание» помогают вам:

- идентифицировать опасность;
- избегать опасности;
- понять последствия.

---

**ВАЖНО**

Идентифицирует информацию, которая критична для реализации приложения и понимания продукта.

---

## Содержание

Данные Logix5000	4
Сервисы CIP	5
Введение	5
Сервис «чтение данных во фрагментированном формате»	14
Сервис CIP «чтение данных»	14
Сервис CIP «запись данных»	15
Сервис «запись данных во фрагментированном формате»	15
Сервис «множественный запрос»	16
Сервис «получить список атрибутов»	17
Сервис «чтение шаблона»	18
Примеры адресации	19
Команды PLC-2	21
Незащищенное чтение	21
Незащищенная запись	22
Защищенная битовая запись	22
Защищенная запись	22
Незащищенная битовая запись	23
Команды PLC-5	24
Чтение-модификация-запись	25
Чтение-модификация-запись N	26
Типовое чтение	27
Чтение диапазона слов	28
Формат ответа	28
Запись диапазона слов	29
Битовая запись	29
Команды SLC	30
SLC защищенное типовое логическое чтение с 3 полями адреса	31
SLC защищенная типовая логическая запись с 3 полями адреса	31
SLC защищенное типовое логическое чтение с 2 полями адреса	32
SLC защищенная типовая логическая запись с 2 полями адреса	32
Приложение А: Отображение адреса	33

## Данные Logix5000

В противоположность контроллеру PLC5, который хранит данные в файлах данных, контроллер Logix5000 хранит данные в тэгах. Тэги Logix5000 имеют следующие свойства:

- имя (name), которое идентифицирует данные:
  - длиной до 40 символов;
  - *не включает* номер файла;
- область действия (scope):
  - контроллер (то есть, глобальная), к которой вы имеете прямой доступ;
  - программа (то есть, локальная), к которой вы *не имеете* доступа через внешнее устройство;
- тип данных, который определяет организацию данных.

Контроллер Logix5000 поддерживает широкий ряд типов данных:

- **простые (atomic)**: бит, байт, 16-битовое слово или 32-битовое слово, каждый из которых хранит единственную величину.
- **структура (structure)**: группа различных типов данных, которые функционируют как одно целое и служат специфическому назначению. В зависимости от потребностей вашего приложения, вы можете создать дополнительные структуры, которые называются структурами, определяемыми пользователем.
- **массив (array)**: последовательность элементов, каждый из которых одного и того же типа данных:
  - вы можете определить размерность ноль, один, два или три, если необходимо. (Размерности ноль и один – наиболее распространены).
  - вы можете использовать атомные или структурные типы данных.

Следующая таблица идентифицирует предопределенные типы данных:

Чтобы сохранить:	Используйте этот тип данных:	Структура
Бит	BOOL	
Битовый массив	BOOLEAN ARRAY (32 bit chunks)	
8-битовое целое	SINT	
16-битовое целое	INT	
32-битовое целое	DINT	
32-битовое с плавающей запятой	REAL	
32 бита миллисекунд	TIMER	✓
32 бита диапазона	COUNTER	✓
32 бита диапазона	CONTROL	✓

## Сервисы СІР

Последующие разделы описывают присущие контроллеру Logix5000 способ связи и адресацию. Для доступа к данным Logix5000 вы можете использовать следующие сервисы:

- сервис СІР – «чтение данных»;
- сервис «чтение данных во фрагментированном формате»;
- сервис СІР – «запись данных»;
- сервис «запись данных во фрагментированном формате»;
- сервис «множественный запрос»;
- сервис «получить список атрибутов»;
- сервис «чтение шаблона».

### Введение

Прежде, чем вы будете использовать сервисы СІР, просмотрите следующую вводную информацию:

- типы данных СІР;
- сегменты ІОІ;
- сокращенный (abbreviated) тип;
- возвращенные форматы данных с сокращенными типами;
- индексация массива;
- ошибки сервиса СІР.

### Типы данных CIP

Информация о типе данных очень важна во всех аспектах связи CIP. Информация типа используется для чтения, записи, и, если необходимо, расшифровывания структуры. Следующая диаграмма показывает формат типов данных CIP.



Структура хранит группу различных типов данных, которые функционируют как единое целое и служат специфической цели (то есть, набор величин).

- Структура содержит один или более членов.
- Каждый элемент может быть:
  - атомным типом данных;
  - другим структурированным типом данных;
  - массивом из данных атомного или структурированного типа.



Чтобы создать тэг, основанный на структуре, у вас есть следующие варианты:

- используйте одну из следующих predefined структур:
  - AXIS,
  - CONTROL,
  - COUNTER,
  - MESSAGE,
  - MOTION\_GROUP,
  - MOTION\_INSTRUCTION,
  - PID,
  - TIMER;
- создайте вашу собственную структуру (тип данных, определяемый пользователем).

Вы можете группировать большинство структур в массивы или использовать их в своих структурах. Вы *не можете* использовать следующие структуры в массивах или своих структурах:

- AXIS,
- MESSAGE,
- MOTION\_GROUP.

### IOI Segments

Адресация CIP выполняется с помощью сегментов IOI. Контроллер использует следующие типы сегментов (придерживаясь Международной спецификации ControlNet):

- версии сегмента «класс»:
  - однобайтовая версия

20	xx
----	----

где:

xx – значение кода класса

- двухбайтовая версия

21	00	xx	xx
----	----	----	----

где:

xxxx – значение кода класса, начиная с младшего байта.



- версии сегмента «событие»:

- однобайтовая версия

24	xx
----	----

где:

xx – значение кода события

- двухбайтовая версия

25	00	xx	xx
----	----	----	----

где:

xx – значение кода события, начиная с младшего байта.

- версии сегмента «элемент»:

- однобайтовая версия

28	xx
----	----

где:

xx – номер элемента

- двухбайтовая версия

29	00	xx	xx
----	----	----	----

где:

xx – номер элемента, начиная с младшего байта.

- четырехбайтовая версия

2A	00	xx младший	xx	xx	xx старший
----	----	---------------	----	----	---------------

где:

xx – номер элемента, с младшего байта по старший..

- Сегмент «символьный», однобайтовая версия:

91	<i>len</i>	<i>1-ый</i>	<i>2-ой</i>	<i>...</i>	<i>последний</i>	<i>00 (дополнение)</i>
----	------------	-------------	-------------	------------	------------------	------------------------

где:

*len* – длина символьной строки, в байтах (исключая заголовок и символ дополнения).

1-ый, 2-ой, 3-ий – символы по порядку.

Дополнение необходимо при нечетном числе символов.

### Сокращенный тип

Чтение и запись таблиц с помощью СІР требует поле сокращенного типа. Это поле идентифицирует тип данных:

- Для данных атомного типа это один байт, являющийся кодом типа, дополненный байтом 0:

<i>xx</i>	<i>00</i>
-----------	-----------

где:

*xx* – величина для таблицы типа данных.

- Для структур:
  - Первый байт указывает, что следует структурный тип.
  - Второй байт обозначает дескриптор структуры, который идентифицирует структуру для целей связи.

<i>A0</i>	<i>02</i>	<i>xx</i>	<i>yy</i>
-----------	-----------	-----------	-----------

где:

*xx yy* – дескриптор структуры из объекта шаблона.

### Возвращенные форматы данных с сокращенными типами

Нижеследующее показывает формат данных и сокращенный тип, возвращенный для всех атомных типов и некоторых из предопределенных структур для обмена CIP сообщениями.

- BOOL

C1	00	xx
----	----	----

где:

xx – 0, если бит равен 0 и FF если бит равен 1.

- BIT ARRAY

D3	00	наимладший	младший	старший	наистарший
----	----	------------	---------	---------	------------

4 байта, упакованных с самого младшего по самый старший.

- SINT

C2	00	xx
----	----	----

где:

xx – значение данных.

- INT

C3	00	младший	старший
----	----	---------	---------

Байты данных, упакованные с младшего по старший.

- DINT

C4	00	наимладший	младший	старший	наистарший
----	----	------------	---------	---------	------------

4 байта, упакованных с самого младшего по самый старший.

- REAL

CA	00	наимладший	младший	старший	наистарший
----	----	------------	---------	---------	------------

4 байта, упакованных с самого младшего по самый старший.

- TIMER

A0	02	??	??	ctl	ctl	ctl	ctl	pre	pre	pre	pre	acc	acc	acc	acc
----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Все слова (ctl, pre и acc) упакованы с самого младшего байта по самый старший байт.

- COUNTER

A0	02	??	??	ctl	ctl	ctl	ctl	pre	pre	pre	pre	acc	acc	acc	acc
----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Все слова (ctl, pre и acc) упакованы с самого младшего байта по самый старший байт.

- CONTROL

A0	02	??	??	ctl	ctl	ctl	ctl	len	len	len	len	pos	pos	pos	pos
----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Все слова (ctl, len и pos) упакованы с самого младшего байта по самый старший байт.

### Индексация массива

Следующие примеры показывают как индексируются двух- и трехмерные массивы, с первого элемента по последний элемент. Все индексы массива начинаются с 0.

Два измерения (3,4)

0	0
0	1
0	2
0	3
1	0
1	1
1	2
1	3
2	0
2	1
2	2
2	3

Три измерения (2,3,2)

0	0	0
0	0	1
0	1	0
0	1	1
0	2	0
0	2	1
1	0	0
1	0	1
1	1	0
1	1	1
1	2	0
1	2	1

## Ошибки сервиса SIP

В общих чертах, состояние SIP возвращается следующим образом:

- нормальное состояние;
- общий код ошибки без расширенного статуса:

код	00
-----	----

Where:	Means:
04	Этот общий код состояния означает, что IOI не может быть расшифрован. Или он не был правильно сформирован или соответствующий тэг не существует.
05	Конкретный пункт на который осуществляется ссылка (обычно событие) не может быть найден.
06	Количество запрошенных данных не помещается в буфер ответа. Произошла частичная передача данных.
0A	Произошла ошибка при попытке обрабатывать один из атрибутов
13	Не достаточно командных данных / параметров были поставлены в команде, чтобы выполнить запрошенную услугу
1C	Предоставлено недостаточное количество атрибутов по сравнению с общим числом атрибутов.
26	Длина слова IOI не соответствовала значению IOI, который был обслужен.
00	УСПЕШНО!!

- ошибка с расширенным статусом:

код	01	расширенный	код
-----	----	-------------	-----

Where:	Means:
FF 01 05 21	Вы попытались достучаться за конец объекта данных.
FF 01 07 21	Сокращение типа не соответствует типу данных объекта данных.
FF 01 04 21	Смещение начала было за концом шаблона.

### Сервис CIP «чтение данных»

Сервис CIP «чтение данных» читает блок данных начиная с адреса, определенного в строке IOI:

- Возвращаются все данные, которые вмещаются в пакет ответа, даже если они не вмещаются в него целиком.
- Если все данные не вмещаются в пакет, то вместе с данными возвращается ошибка.

#### Формат команды

Код сервиса 4C	Строка IOI	размер
-------------------	------------	--------

где:

*размер* – количество элементов, которые вы хотите читать (16 битовых).

#### Формат ответа

Код сервиса CC	00	статус	сокращенный тип	данные
-------------------	----	--------	--------------------	--------

### Сервис «чтение данных во фрагментированном формате»

Этот сервис читает данных больше, чем вмещается в один пакет:

- Приложение должно делить данные на блоки соответствующего размера.
- Клиентский процесс должен, затем, запрашивать последующий части данных по очереди.
- Единственный параметр, который отличает один запрос от последующего – смещение байта (32 бита, начиная с наименее значимого бита).
- С каждым пакетом кроме последнего, возвращается ошибка размера.

#### Формат команды

Код сервиса 52H	Строка IOI	размер	смещение байта
--------------------	------------	--------	-------------------

где:

*размер* – количество элементов, которые вы хотите читать (16 битовых).

*Формат ответа*

Код сервиса D2H	00	статус	сокращенный тип	данные
--------------------	----	--------	--------------------	--------

**Сервис CIP «запись данных»**

Сервис CIP «запись данных» пишет блок данных начиная с адреса, определенного в строке IOI. Чтобы запись произошла, типы данных должны точно соответствовать.

*Формат команды*

Код сервиса 4D	IOI	сокращенный тип	размер	данные
-------------------	-----	--------------------	--------	--------

*Формат ответа*

Код сервиса CD	00	статус
-------------------	----	--------

**Сервис «запись данных во фрагментированном формате»**

Этот сервис записывает данных больше, чем вмещается в один пакет:

- Приложение должно делить данные на блоки соответствующего размера.
- Клиентский процесс должен, затем, посылать последующий части данных по очереди.
- Единственный параметр, который отличает одну запись от последующей – смещение байта (32 бита, начиная с наименее значимого бита).

*Формат команды*

Код сервиса 53H	IOI	сокращенный тип	размер	смещение байта	данные
--------------------	-----	--------------------	--------	-------------------	--------

*Формат ответа*

Код сервиса D3H	00	статус
--------------------	----	--------

### Сервис «множественный запрос»

Сервис «множественный запрос» упаковывает в пакет более, чем один запрос. Используйте этот сервис, чтобы оптимизировать чтение и запись СІР. Чтение или запись только одного тэга за один раз может требовать весьма значительного времени. Большая часть времени затрачивается на доставку сообщения контроллер и назад, а не сколько-нибудь значимое время обработки данных любой из сторон.

#### Формат команды

0A	02	20	02	24	01	счетчик
смещение #1	смещение #2	...	сервис #1	сервис #2	...	сервис #n

*счетчик* – двухбайтовое поле количества сервисов для выполнения.

*смещение #n* – двухбайтовое поле, которое дает байтовое смещение (от поля счетчика) на начало *n*-ного сервиса.

*сервис #n* – один из сервисов для выполнения. Примеры форматов для чтения и записи СІР показаны выше.

В любом случае, сервисы и вся другая заголовочная информация должны уместиться в пределах границ пакета.

#### Формат ответа

8A	00	Gen STS	00	счетчик	смещение #1
смещение #2	...	отклик #1	отклик #2	...	отклик #n

где:

*счетчик* – двухбайтовое поле количества откликов в ответе.

*смещение #n* – двухбайтовое поле, которое дает байтовое смещение (от поля счетчика) на начало *n*-ного отклика.

*отклик #n* – отклик на *n*-ный сервис в запросе с последующим ответом в одном из форматов чтения и записи СІР, показанных выше.

Всегда обрабатываются все сервисы. Каждый отклик содержит статус для своего собственного сервиса. Байт Gen STS указывает на то, что при обработке сервисов где-то произошла ошибка.



## Сервис «получить список атрибутов»

Этот сервис возвращает важную информацию о шаблоне:

- размер структуры;
- количество байтов;
- количество членов;
- дескриптор.

Могут потребоваться множественные чтения, чтобы получать все данные шаблона.

### Формат команды

03	03	20	6C	25	00	##	##	04	00	04	00	03	00	02	00	01	00
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

где:

*## ##* – номер события для шаблона, который вы читаете, начиная с младшего байта.

### Формат ответа

83	00	00	00	04	00	04	00	00	00	ss	ss	ss	ss	03	00
00	00	bb	bb	02	00	00	00	mm	mm	01	00	00	00	hh	hh

где:

*ss ss ss ss* – размер структуры шаблона в 32 битовых словах.

*bb bb* – количество байтов, которое структура занимает в памяти.

*mm mm* – количество членов в структуре.

*hh hh* – дескриптор структуры используемый для связи.

### Сервис «чтение шаблона»

Этот сервис возвращает информацию содержащуюся в шаблоне о типе каждого из элементов и где он начинается в структуре (байтовое смещение). Может потребоваться много чтений, чтобы получить все данные шаблона.

#### Формат команды

4C	03	20	6C	25	00	tt	tt	rr	rr	rr	rr	bb	bb
----	----	----	----	----	----	----	----	----	----	----	----	----	----

где:

*tt tt* – номер события для шаблона, который вы читаете, начиная с младшего байта.

*rr rr rr rr* – байтовое смещение начала чтения. Для чтения первого байта смещение – 0.

*bb bb* – количество байтов для чтения.

#### Формат ответа

CC	00	00	00	данные
----	----	----	----	--------

## Примеры адресации

Следующие примеры показывают строки IOI для доступа к различным данным:

- в строках IOI первый байт всегда является счетчиком количества слов;
- в некоторых случаях также показана логическая форма PCCC ASCII (предполагает отображение (mapping) файлов).

Примеры сгруппированы в следующие секции:

- доступ к атомным и предопределенным типам данных – на странице 19;
- доступ к структурам, определенным пользователем – на странице 20.

### Доступ к атомным и предопределенным типам данных

Следующие примеры доступны к тэгам, которые используют атомный или предопределенный тип данных.

Для доступа к:	Определен следующий вход:	Примечания:
одному INTEGER тэгу, названному "parts"	IOI строка 04 91 05 70 61 72 74 73 00	
	PCCC символичный "parts"	
6-му элементу массива REAL, названному "setpoints"	IOI строка 07 91 09 73 65 74 70 6f 69 7e 74 73 00 28 05	
	PCCC ASCII \$F8:6	
	PCCC символичный "setpoints(6)"	
одному INTEGER (2,5,257) из трехмерного массива, названного "profile"	IOI строка 09 91 07 70 72 6f 66 69 6c 65 00 28 02 28 05 29 00 01 01	
	PCCC ASCII \$N7:2:5:257	
	PCCC символичный "profile(2,5,257)"	
одному тэгу таймера, названному "dwell3"	IOI строка 04 91 06 64 77 65 6c 6c 33	Возвращает 12 байтов данных
	PCCC символичный "dwell3"	
двухмерному массиву счетчиков (5,0), названному "counts"	IOI строка 06 91 06 63 6f 75 6e 74 73 28 05 28 00	Требуется последнее измерение со значением 0
	PCCC ASCII \$C5:5:0	
	PCCC символичный "counts(5,0)"	
накопленному значению таймера, названного "dwell3"	IOI строка 07 91 06 64 77 65 6c 6c 33 91 03 61 63 63 00	Добавьте строку "асс" к IOI как другой символичный сегмент.
	PCCC ASCII \$T4:0:2	".ACC" не работает
величина уставки двухмерного массива счетчиков (5,0), названных "counts"	IOI строка 09 91 06 63 6f 75 6e 74 73 28 05 28 00 91 03 70 72 65 00	
	PCCC ASCII \$C5:5:0:1	".PRE" не работает

## Доступ к структурам, определенным пользователем

Следующий набор примеров достигается к следующим структурам, определенным пользователем:

Имя структуры: STRUCT_A		Имя структуры: STRUCT_B	
Член:	Тип данных:	Член:	Тип данных:
limit4	Бит	pilot_on	Бит
limit7	Бит	hourlyCount	INT(12)
travel	DINT	rate	REAL
errors	DINT		
wear	REAL		

Имя структуры: STRUCT_C		Имя структуры: STRUCT_D	
Член:	Тип данных:	Член:	Тип данных:
hours_full	Бит	myint	INT
today	STRUCT_B	myfloat	REAL
sampleTime	TIMER	myarray	STRUCT_C(8)
shipped	COUNTER	mypid	PID

Чтобы иметь доступ к:	Определена следующая строка IOI :
одиночному тэгу STRUCT_A "struct1"	05 91 07 73 74 72 75 63 74 31 00
члену "wear" в одиночной структуре STRUCT_A "struct1"	08 91 07 73 74 72 75 63 74 31 00 91 04 77 65 61 72
члену "travel" в одномерном массиве (9) структур STRUCT_A "str1Array"	0B 91 09 73 74 72 31 41 72 72 61 79 00 28 09 91 06 74 72 61 76 65 6c
5-му элементу "hourlyCount" одиночной структуры STRUCT_B "struct2"	0D 91 07 73 74 72 75 63 74 32 00 91 0B 68 6f 75 72 6c 79 43 6f 75 6e 74 00 28 05
элементу "rate" STRUCT_B в структуре STRUCT_C "struct3"	0C 91 07 73 74 72 75 63 74 33 00 91 05 74 6f 64 61 79 00 91 04 72 61 74 65
STRUCT_D создана с двумя измерениями "my2Dstruct4"	15 91 0B 6d 79 32 44 73 74 72 75 63 74 34 00 28 04 28 05 91 07 6d 79 61 72 72 61 79 00 28 04 91 0A 73
TIMER'у 4-ой STRUCT_C в STRUCT_D "my2Dstruct4", элемент (4,5)	61 6d 70 6c 65 54 69 6d 65
3-му INTEGER члену "hourlyCount" STRUCT_B в	1B 91 0B 6d 79 32 44 73 74 72 75 63 74 34 00 28 03
6-ом члене "myarray" STRUCT_C 2-мерного массива STRUCT_D "my2Dstruct4", элемент (3,2)	28 02 91 07 6d 79 61 72 72 61 79 00 28 06 91 05 74 6f 64 61 79 00 91 0B 68 6f 75 72 6c 79 43 6f 75 6e 74 00 28 03

## Команды PLC-2

Используйте команды PLC-2, для доступа к одному тэгу (только одному тэгу) в контроллере Logix5000. Дополнительно к посылке команды, вы должны также отобразить (map) сообщение на тэг типа INT (16 битовое целое) в контроллере Logix5000. Смотрите «Приложение А: Отображение адреса» на странице 33.

Контроллеры Logix5000 поддерживают следующие команды PLC-2:

- незащищенное чтение;
- защищенная запись;
- незащищенная запись;
- защищенная битовая запись;
- незащищенная битовая запись.

### Незащищенное чтение

Эта команда обеспечивает возможность чтения для команд PLC-2.

#### Формат команды

CMD	STS	TNS	TNS	адрес	смещение	размер
-----	-----	-----	-----	-------	----------	--------

где:

*смещение адреса* должно находиться на 16 битовой границе;

*размер* должен быть четным количеством байтов.

#### Формат ответа

CMD	STS	TNS	TNS	данные
-----	-----	-----	-----	--------

где:

*данные* вплоть до 244 байтов.

### Защищенная запись

Эта команда обеспечивает возможность защищенной записи для команд PLC-2.

Формат команды

CMD	STS	TNS	TNS	адрес	смещение	данные
-----	-----	-----	-----	-------	----------	--------

Формат ответа

CMD	STS	TNS	TNS
-----	-----	-----	-----

### Незащищенная запись

Эта команда обеспечивает базовую возможность записи для команд PLC-2

Формат команды

CMD	STS	TNS	TNS	адрес	смещение	данные
-----	-----	-----	-----	-------	----------	--------

Формат ответа

CMD	STS	TNS	TNS
-----	-----	-----	-----

### Защищенная битовая запись

Эта команда обеспечивает возможность защищенной битовой записи для команд PLC-2.

Формат команды

				← повторяемо, вплоть до 61 раза →		
CMD 02	STS	TNS	TNS	PLC-2 адрес	маска set	маска reset

где:

*адрес PLC-2* – обычное двухбайтовое смещение;

*маска set* – один байт;

*маска reset* – один байт.

Формат ответа

CMD 42	STS	TNS	TNS
-----------	-----	-----	-----

### Незащищенная битовая запись

Эта команда обеспечивает возможность битовой записи для команд PLC-2.

Формат команды

				← повторяемо, вплоть до 61 раза →		
CMD 05	STS	TNS	TNS	PLC-2 адрес	<i>маска set</i>	<i>маска reset</i>

где:

*адрес PLC-2* – обычное двухбайтовое смещение;

*маска set* – один байт;

*маска reset* – один байт.

Формат ответа

CMD 45	STS	TNS	TNS
-----------	-----	-----	-----

## Команды PLC-5

Каждая команда PLC-5 требует системный адрес в одной из следующих форм:

- логическая двоичная или логическая ASCII, которая адресует данные как «файл», «элемент» и т.д.:
  - Первый уровень логического двоичного адреса всегда должен быть 0. Это требуется для доступа к тэгам уровня контроллера (controller-scoped).
  - Второй уровень – номер "файла". Это тот уровень, который следует за буквой (буквами) в логической форме ASCII.
  - Следующие 1, 2 или 3 уровня соответствуют индексам измерений массива следующим образом: data[1][2][3].
  - Любые последующие уровни логического адреса доступны к элементам сложных типов. Обратитесь к «Типам данных CIP» на странице 6.

Обратитесь к «Приложению A: Отображение адресов» на странице 33.

- символьная, которая адресует данные непосредственно по имени тэга:
  - Символьная строка начинающаяся и заканчивающаяся символом NULL.
  - В самом простом случае, символьная строка состоит из имени тэга.
  - Для адресации массива разделите индексы массива квадратными скобками.

Следующие примеры показывают символическую адресацию.

---

**ПРИМЕР**

Символические адреса

- tag\_name
- tag\_name[x]
- tag\_name[x,y,z]
- tag\_name[x][y][z].

---

Контроллеры Logix5000 поддерживают следующие команды PLC-5:

- чтение-модификация-запись;
- чтение-модификация-запись N;
- типовое чтение;
- типовая запись;
- чтение диапазона слов;
- запись диапазона слов;
- битовая запись.



**ВАЖНО**

Для команд типового чтения и записи, вы можете иметь доступ только к данным следующих типов:

- SINT
- INT
- DINT
- REAL

**Чтение-модификация-запись**

Поле системного адреса PLC-5 определяет слово, которое должно быть модифицировано. За каждым системным адресом PLC-5 следует поле маски И и поле маски ИЛИ (по два байта каждый, начиная с младшего). Вы можете использовать более, чем один набор таких полей, чтобы указать на модификацию более чем одного слова.

Для каждого из таких наборов трех полей (адрес, маска И, маска ИЛИ), выполняется следующая процедура:

1. Копируется указанное слово.
2. Сбрасываются биты, указанные в маске И.
3. Устанавливаются биты, указанные в маске ИЛИ.
4. Слово записывается назад.

**Формат команды**

					← повторяемо, вплоть до 243 байт →				
CMD OF	STS	TNS	TNS	FNC 26	PLC-5 си- стемный адрес	И	маска	ИЛИ	маска

где:

*маска И* – определяет какие биты в слове сбросить (0);

*маска ИЛИ* – определяет какие биты в слове установить (1).

## Формат ответа

CMD 4F	STS	TNS	TNS	EXT STS
-----------	-----	-----	-----	------------

**Чтение-модификация-запись N**

Системный адрес PLC-5 определяет какое слово будет модифицировано. За каждым адресом следует поле длины. Применимые значения – 2 и 4 байта. Маски И и ИЛИ следует за длиной области, соответственно. Эти четыре области могут повторяться вплоть до длины в 243 байта.

Для каждого из таких наборов четырех полей (адрес, длина маски, маска И, маска ИЛИ), выполняется следующая процедура:

1. Копируется указанное слово.
2. Сбрасываются биты, указанные в маске И.
3. Устанавливаются биты, указанные в маске ИЛИ.
4. Слово записывается назад.

## Формат команды

						← повторяемо, вплоть до 243 байт →			
CMD 0F	STS	TNS	TNS	FNC 79	число наборов	PLC-5 си- стемный адрес	длина маски	И маска	ИЛИ маска

где:

*маска И* – определяет какие биты в слове сбросить (0);

*маска ИЛИ* – определяет какие биты в слове установить (1).

## Формат ответа

Повторяется для каждого набора, старшее слово если необходимо

EXT STS		Отклик		
------------	--	--------	--	--

### Типовое чтение

Команда типового чтения считывает блок данных из контроллера, начиная с системного адреса PLC-5 плюс смещение пакета.

#### Формат команды

CMD 0F	STS	TNS	TNS	FNC 68	пакета	сме- щение	размер	тран- закции	PLC-5 си- стемный адрес	размер
-----------	-----	-----	-----	-----------	--------	---------------	--------	-----------------	-------------------------------	--------

где:

*размер* – количество элементов для чтения.

#### Формат ответа

CMD 4F	STS	TNS	TNS	<i>a</i>	<i>b</i>
-----------	-----	-----	-----	----------	----------

где:

*a* – отображает информацию о типе или расширенный статус при ошибке команды;

*b* – байты данных.

#### Типовая запись

Команда типовой записи прописывает блок данных в контроллер, начиная с системного адреса PLC-5 плюс смещение пакета.

#### Формат команды

CMD 0F	STS	TNS	TNS	FNC 67	пакета	сме- щение	размер	тран- закции	PLC-5 сист. адрес	<i>a</i>	<i>b</i>
-----------	-----	-----	-----	-----------	--------	---------------	--------	-----------------	-------------------------	----------	----------

где:

*смещение пакета* – в количестве элементов;

*размер транзакции* – в количестве элементов;

*a* – указывает количество элементов для записи;

*b* – байт данных.

### Формат ответа

Расширенный статус существует только если есть ошибка.

CMD 4F	STS	TNS	TNS	EXT STS
-----------	-----	-----	-----	------------

### Чтение диапазона слов

Команда чтения диапазона слов считывает блок слов из контроллера, начиная с системного адреса PLC-5 плюс смещение слова.

### Формат команды

CMD 0F	STS	TNS	TNS	FNC 01	<i>пакета</i>	<i>сме- щение</i>	<i>размер</i>	<i>тран- закции</i>	<i>PLC-5 си- стемный адрес</i>	<i>размер</i>
-----------	-----	-----	-----	-----------	---------------	-----------------------	---------------	-------------------------	--	---------------

где:

*смещение пакета* – в количестве 16 битовых слов;

*размер транзакции* – в количестве 16 битовых слов;

*размер* – в количестве байтов и должно быть четным.

### Формат ответа

CMD 0F	STS	TNS	TNS	<i>данные</i>
-----------	-----	-----	-----	---------------

где:

*данные* – вплоть до 244 байтов.

### Запись диапазона слов

Команда записи диапазона слов пропишет блок данных в контроллер, начиная с системного адреса PLC-5 плюс смещение слова.

Формат команды

CMD 0F	STS	TNS	TNS	FNC 00	пакета	сме- щение	размер	тран- закции	PLC-5 си- стемный адрес	данные
-----------	-----	-----	-----	-----------	--------	---------------	--------	-----------------	-------------------------------	--------

где:

*смещение пакета* – в количестве 16 битовых слов;

*размер транзакции* – в количестве 16 битовых слов.

Формат ответа

CMD 4F	STS	TNS	TNS	EXT STS
-----------	-----	-----	-----	------------

### Битовая запись

Эта команда установит и сбросит биты в единственном слове определенном логическим адресом PLC-5. Любой данной командой можно изменить только единственное слово.

Формат команды

CMD 0F	STS	TNS	TNS	FNC 02	PLC-5 си- стемный адрес	set	маска	reset	маска
-----------	-----	-----	-----	-----------	-------------------------------	-----	-------	-------	-------

Формат ответа

CMD 4F	STS	TNS	TNS	EXT STS
-----------	-----	-----	-----	------------

## Команды SLC

Команды SLC используют строго логическую форму адресации (то есть – файл / элемент / подэлемент). Смотрите «Приложение А: Отображение адреса» на странице 33.

Логическая адресация SLC имеет ограниченное число уровней логического адреса, поэтому существуют некоторые специальные моменты, на которые следует обратить внимание:

<b>В:</b>	<b>Номер элемента используется, как:</b>
одномерном массиве	индекс измерения для адресации (данные(elem))
двухмерном массиве	индекс второго измерения, а первый индекс измерения всегда равен 0 (данные(0)(elem))
трехмерном массиве	индекс третьего измерения, а индексы первого и второго измерения равны 0 (данные(0)(0)(elem))

Применимы следующие ограничения:

- Вы можете использовать только файловые типы SLC. Для информации о типах данных SLC, обратитесь «Справочнику набору команд и протоколу DF1», публикация 1770-6.5.16.
- Для команд типового чтения и записи, вы можете иметь доступ только к следующим типам данных:
  - SINT;
  - INT;
  - DINT;
  - REAL.

Контроллеры Logix5000 поддерживают следующие команды SLC:

- SLC защищенное типовое логическое чтение с 3 полями адреса;
- SLC защищенная типовая логическая запись с 3 полями адреса;
- SLC защищенное типовое логическое чтение с 2 полями адреса;
- SLC защищенная типовая логическая запись с 2 полями адреса.

**SLC защищенное типовое логическое чтение с 3 полями адреса**

Сервис поддерживается для совместимости с модулями SLC.

*Формат команды*

CMD 0F	STS	TNS	TNS	FNC A2	<i>размер (байт)</i>	<i>номер файла</i>	<i>тип файла</i>	<i>номер эле- мента</i>	<i>номер подэле- мента</i>
-----------	-----	-----	-----	-----------	--------------------------	------------------------	----------------------	---------------------------------	------------------------------------

*Формат ответа*

CMD 4F	STS	TNS	TNS	EXT STS	<i>данные</i>
-----------	-----	-----	-----	------------	---------------

**SLC защищенная типовая логическая запись с 3 полями адреса**

Этот сервис поддерживается для совместимости со старыми модулями.

*Формат команды*

CMD 0F	STS	TNS	TNS	FNC AA	<i>размер (байт)</i>	<i>номер файла</i>	<i>тип файла</i>	<i>номер эле- мента</i>	<i>номер подэле- мента</i>	<i>данные</i>
-----------	-----	-----	-----	-----------	--------------------------	------------------------	----------------------	---------------------------------	------------------------------------	---------------

*Формат ответа*

CMD 4F	STS	TNS	TNS	EXT STS
-----------	-----	-----	-----	------------

**SLC защищенное типовое логическое чтение с 2 полями адреса**

Эта команда чтения обеспечивает простую версию чтения данных.

*Формат команды*

CMD 0F	STS	TNS	TNS	FNC A1	<i>размер (байт)</i>	<i>номер файла</i>	<i>тип файла</i>	<i>номер эле- мента</i>
-----------	-----	-----	-----	-----------	--------------------------	------------------------	----------------------	---------------------------------

*Формат ответа*

CMD 4F	STS	TNS	TNS	EXT STS	<i>данные</i>
-----------	-----	-----	-----	------------	---------------

**SLC защищенная типовая логическая запись с 2 полями адреса**

Эта команда записи обеспечивает простую версию записи данных.

*Формат команды*

CMD 0F	STS	TNS	TNS	FNC A9	<i>размер (байт)</i>	<i>номер файла</i>	<i>тип файла</i>	<i>номер эле- мента</i>	<i>данные</i>
-----------	-----	-----	-----	-----------	--------------------------	------------------------	----------------------	---------------------------------	---------------

*Формат ответа*

CMD 4F	STS	TNS	TNS	EXT STS
-----------	-----	-----	-----	------------



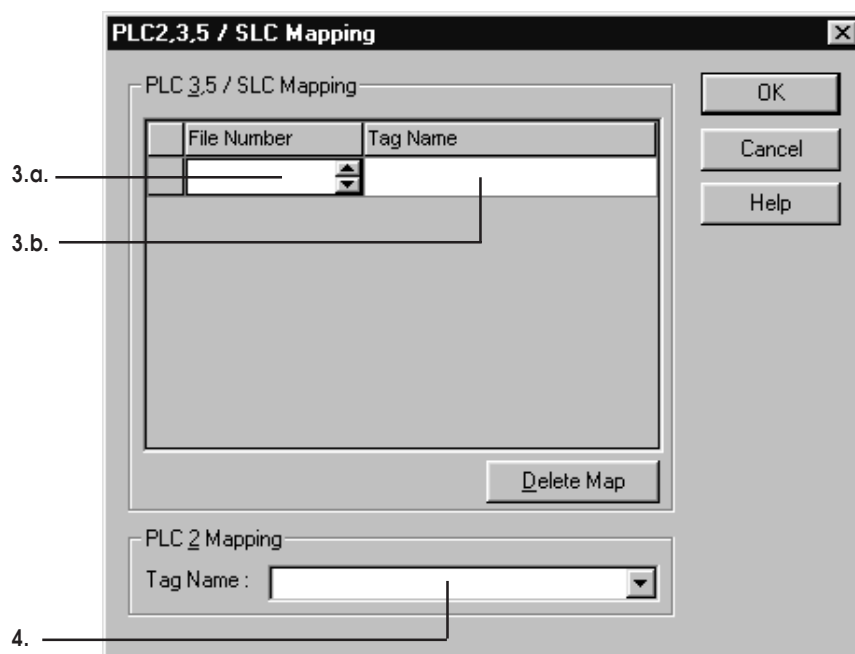
## Приложение А: Отображение адреса

Чтобы иметь доступ к данным контроллера Logix5000, используя команды PLC или SLC, вы должны отобразить тэги на файлы:

- Вы должны отобразить только номера файлов, которые используются в сообщениях; нет необходимости отображать другие номера файлов.
- Таблица отображения загружается в контроллер и используется всякий раз, когда имеет место доступ к данным с "логической" адресацией.
- Вы можете иметь доступ только к тэгам на уровне контроллера (глобальным данным).

Чтобы отобразить адрес:

1. В программном обеспечении RSLogix 5000, откройте файл проекта для контроллера, доступ к данным которого вы хотите иметь.
2. Из меню **Logic**, выберите **Map PLC/SLC Messages**.



3. Для каждого файла, который указывается в команде PLC-5 или SLC, сделайте вход в карту отображения:
  - a. Наберите номер файла логического адреса.
  - b. Наберите или выберите тэг уровня контроллера (глобальный), который будет источником или приемником данных для этого номера файла. (Вы можете отобразить один тэг на несколько файлов).
4. Для команд PLC-2, определите тэг, который будет источником или приемником данных.
5. Щелкните **OK**.

**Примечания:**