Программируем контроллер модульной линейки FASTWEL I/O CPM713

Светлана Захаркина, Анастасия Казначеева, Александр Локотков

В статье приводятся ответы на часто задаваемые вопросы пользователей системы FASTWEL I/O CPM713. Описываются готовые решения в области подключения, диагностики и программирования контроллера.

Вопрос

Как оценить уровень загрузки процессора CPM713?

Ответ

Оценить уровень загрузки процессора **CPM713** можно следующими способами.

Способ 1: визуально оценить степень свечения верхних светодиодов RUN/ERR и APP на панели индикаторов (рис. 1). Если светодиод RUN/ERR светится красным цветом, то в приложении имеется более одной циклической задачи и ни одна из них не укладывается в заданный период. Если индикатор светится зелёным цветом, то в приложении имеется единственная циклическая задача и она хотя бы иногда успевает укладываться в заданный период, либо в приложении имеется более одной циклической задачи и хотя бы одна из них хотя бы иногда укладывается в заданный период. Прерывистое свечение красным цветом индикатора APP означает, что все циклические задачи никогда не успевают укладываться в заданный период и хотя бы одна циклическая задача иногда успевает укладываться в заданный период. Зелёный цвет индикатора APP означает, что все циклические задачи всегда успевают укладываться в заданный период.

Способ 2: программным путём проверить статус битовых полей диагностики исполнения приложения и счётчики циклов и запаздываний (рис. 2).

В конфигурации контроллера имеется секция *Diagnostics–Application*, в которой определены два входных канала, позволяющих приложению во время выполнения получить общее количество циклов всех циклических задач и общее количество циклов, во время которых циклические задачи не успели завершить исполнение в течение заданных периодов. Назначение каналов представлено в табл. 1.

Способ 3: воспользоваться функцией *F_IecTasks_getInfo* из библиотеки *FastwelTasksExchange.lib*. Данная функция принимает указатель на переменную типа *F_TASK_INFO* в качестве первого параметра и возвращает диагностическую информацию о задаче, номер которой передан вторым параметром. Если задача с данным номером отсутствует в системе, функция возвращает 0.

Структура *F_TASK_INFO* определена следующим образом:

```
TYPE F_TASK_INFO :
STRUCT
period_us : DWORD;
(* период циклической задачи в мкс
или для ациклической задачи –
16#FFFFFFFF *)
```

Рис. 1. Панель индикаторов

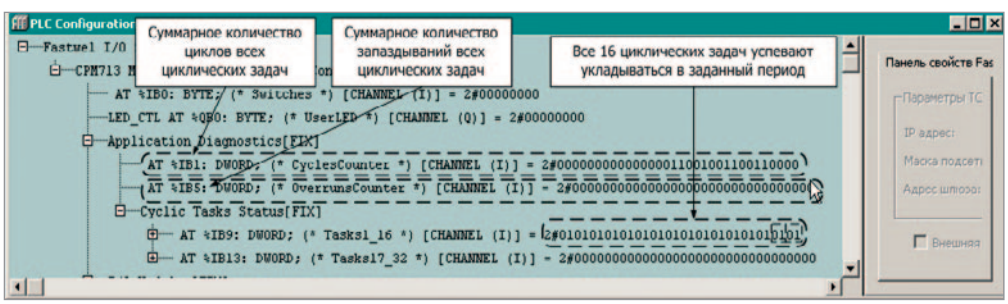


Рис. 2. Диагностические каналы среды исполнения CODESYS

Таблица 1

Описание секции Diagnostics–Application конфигурации контроллера узла

Элемент/канал	Адрес	Тип	Назначение
CyclesCounter	%IB1	DWORD	Общее количество циклов всех циклических задач
OverrunsCounter	%IB5	DWORD	Общее количество циклов циклических задач, во время которых они не успели завершить выполнение в течение своих заданных периодов
Cyclic Tasks Status–Task1_16	%IB9	DWORD	Двухбитовые статусы циклических задач

```
PLC_PRG (PRG-ST)
0001 PROGRAM PLC_PRG
0002 VAR
0003   b1:BLINK;
0004   analog_converter:AIM720_DIRECT;
0005   Task_info:F_TASK_INFO;
0006 END_VAR
0001 b1(enable:=TRUE, timelow:=t#2500ms, timehigh:=t#2500ms, out=>relay_out3);
0002 relay_out1:=relay_net1;
0003 relay_out2:=relay_net2;
0004 analog_converter;
0005 volt_net1:=analog_converter.outputs.iout0;
0006 volt_net2:=analog_converter.outputs.iout1;
0007
0008 F_IecTasks_getInfo(ADR(Task_info),16#FFFF);
0009
```

Рис. 3. Пример использования функции F_IecTasks_getInfo

```
PLC_PRG (PRG-ST)
0003 Task_info
0004     .period_us = 80000
0005     .cyclesCount = 695
0006     .overrunsCount = 0
0007     .minExecutionTime_us = 40
0008     .maxExecutionTime_us = 74
0009     .name = 'NewTask'
0010     .startCycleTickCount_us = 907070686
0011     .lastExecutionTime_us = 45
0012
```

Рис. 4. Результат выполнения программного кода

```
cyclesCount : DWORD;
(* количество циклов, выполненных задачей *)
overrunsCount : DWORD;
(* количество запаздываний циклической задачи *)
minExecutionTime_us : DWORD;
(* минимальное время ввода данных и выполнения пользовательского кода, мкс *)
maxExecutionTime_us : DWORD;
(* максимальное время ввода данных и выполнения пользовательского кода, мкс *)
name : STRING(23);
(* имя задачи *)
startCycleTickCount_us : DWORD;
(* счётчик микросекунд в момент последнего запуска задачи перед вызовом F_IecTasks_getInfo *)
lastExecutionTime_us : DWORD;
(* время ввода данных и выполнения пользовательского кода в мкс в цикле, предшествующем вызову F_IecTasks_getInfo *)
END_STRUCT
END_TYPE
```

Номер задачи, передаваемый в качестве второго параметра, является индексом задачи (начиная с 0) в древовидном списке ресурса *Task Configuration* среды разработки CODESYS.

При вызове *F_IecTasks_getInfo* в контексте какой-либо циклической задачи в качестве номера может использоваться значение 16#FFFF. В этом случае функция вернёт статистику для текущей циклической задачи. Пример программы с использованием функции *F_IecTasks_getInfo* приведён на рис. 3 и 4.

Для рассмотренной в примере задачи:
`period_us = 80000` — период выполнения, мкс;
`cyclesCount = 695` — количество циклов, выполненных задачей;
`overrunsCount = 0` — количество циклов, на которых задача не уложилась в заданный период исполнения;
`minExecutionTime_us = 40` — минимальное время исполнения, мкс;
`maxExecutionTime_us = 74` — максимальное время исполнения, мкс;
`name = 'NewTask'` — имя задачи;
`startCycleTickCount_us = 907070686` — счётчик микросекунд в момент последнего запуска задачи перед вызовом *F_IecTasks_getInfo*;
`lastExecutionTime_us = 45` — время ввода данных и выполнения пользовательского кода в мкс в цикле, предшествующем вызову *F_IecTasks_getInfo*.

Вопрос

Как установить связь с контроллером CPM713 в среде программирования CODESYS V2.3?

Ответ

Если контроллер новый, только что распакован, то соединиться с ним можно по адресу 10.0.0.1 (маска 255.0.0.0). Также с контроллером можно соединиться по интерфейсу «точка—точка» (P2P) через COM-порт с помощью кабеля, входящего в комплект поставки.

В случае если контроллер CPM713 уже использовался ранее, но по каким-то причинам нет информации о его коммуникационных параметрах и отсутствует рабочий проект, загруженный ранее, то для установки связи с контроллером необходимо выполнить следующую последовательность действий.

1. Перевести контроллер в безопасный режим:
   1.1. Перевести первый переключатель в положение ON (вправо).
   1.2. Выключить, а затем включить питание контроллера.
   1.3. Дождаться загрузки контроллера (попеременное свечение индикатора RUN/ERR зелёным и красным цветом).
   1.4. Вернуть первый переключатель в положение OFF (влево), чтобы при следующей перезагрузке питания контроллер не ушёл в безопасный режим.
   1.5. Проверить подключение к контроллеру: IP 10.0.0.1, маска подсети 255.0.0.0.

Если рабочий проект существует, то дополнительно к п. 1 необходимо сделать следующее.

2. Загрузить рабочий проект в контроллер:
   2.1. Открыть проект в среде программирования CODESYS V2.3.
   2.2. На вкладке Ресурсы (Resources) в окне Конфигурация ПЛК (PLC Configuration) в поле Ethernet Multiprotocol Port установить допустимые в используемом сегменте сети параметры со значе-

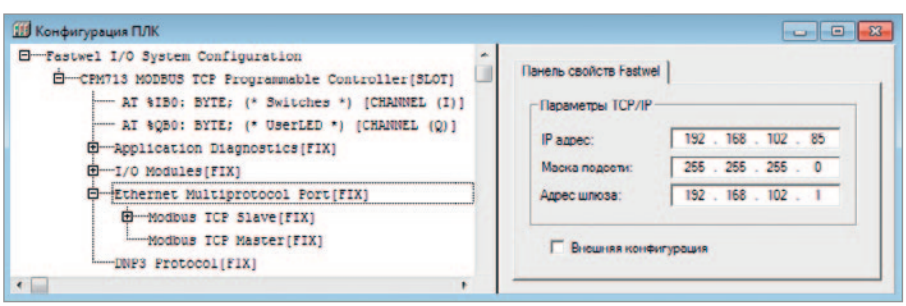


Рис. 5. Окно сетевых параметров

ниями IP-адреса, маски подсети и адреса шлюза (рис. 5).

2.3. Открыть окно Параметры подключения (Communication Parameters) из меню Онлайн (Online). Добавить новое соединение (New) и выбрать пункт Modbus TCP (Fastwel Modbus TCP). По умолчанию в строке Address указывается заводской адрес контроллера 10.0.0.1.

2.4. Загрузить проект в контроллер командой Онлайн (Online)/Подключение (Login).

2.5. После загрузки проекта связь среды с контроллером оборвётся, потому что IP-адрес контроллера изменится на новый.

2.6. Чтобы подключиться ещё раз, необходимо в окне Параметры коммуникации в строке Address изменить адрес контроллера с 10.0.0.1 на нужный.

Вопрос

Можно ли программным способом установить и прочитать пользовательский серийный номер контроллера CPM713 или его MAC-адрес с целью привязать программу к данному контроллеру?

Ответ

Установить, а затем прочитать пользовательский серийный номер контроллера CPM713 можно с помощью функций FwPlatformSetSerialNumber и FwPlatformGetSerialNumber системной библиотеки FastwelPlatformControl.lib. Пример использования этих функций показан на рис. 6 и 7. Установить и прочитать MAC-адрес контроллера невозможно.

Вопрос

Существуют ли у контроллеров CPM713 какие-либо ограничения на запись значений через указатель POINTER в переменные Modbus?

В пояснение вопроса рассмотрим пример кода:

```
pt_write[1]:=ADR(Out_m_Data01);
pt_write[1]^:=Data_out[1];
pt_write[2]:= pt_write[1]+2;
pt_write[2]^:= Data_out[2];
```

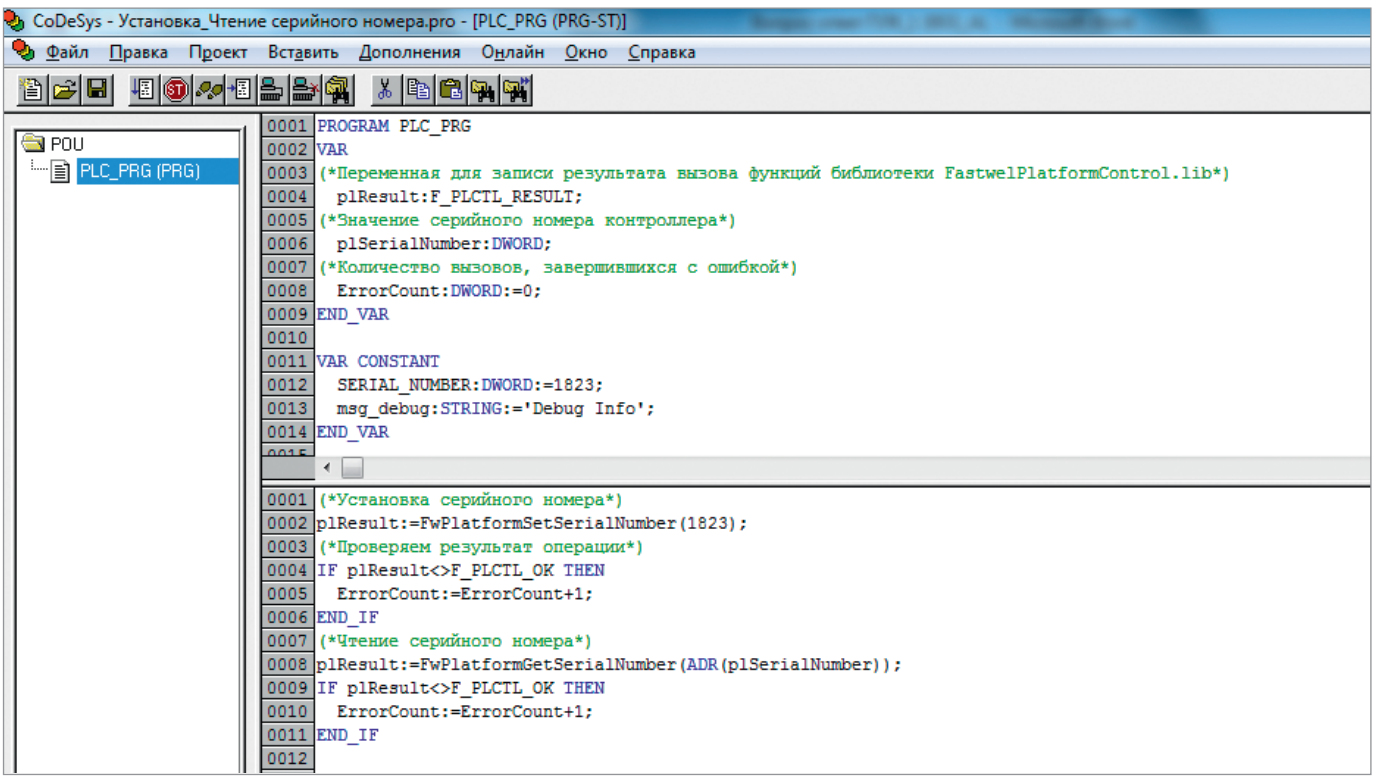


Рис. 6. Пример использования функций библиотеки FastwelPlatformControl.lib

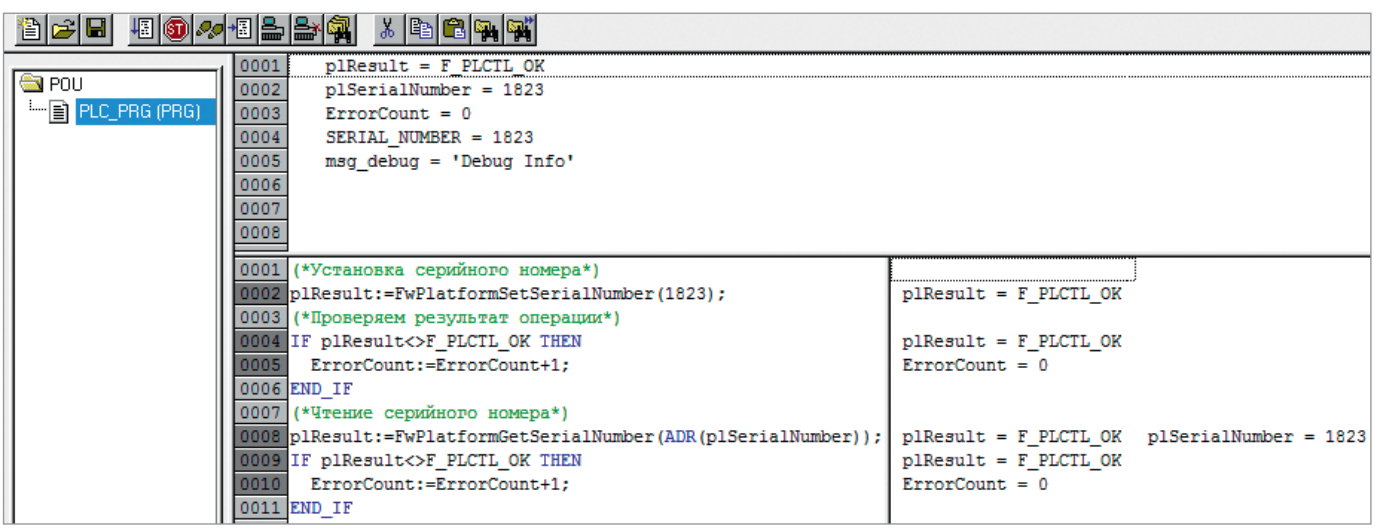


Рис. 7. Результат выполнения программного кода

```
Ethernet Multiprotocol Port[FIX]
  Modbus TCP Slave[FIX]
    AT %IB2577: DWORD; (* TransactionsCount *) [CHANNEL (I)] = 50779
    AT %IB2581: DWORD; (* ErrorsCount *) [CHANNEL (I)] = 5
    Inputs[FIX]
    Outputs[FIX]
      WORD Output[VAR]
        Out_m_Data01 AT %QB2561: WORD; (*  *) [CHANNEL (Q)] = 42
      WORD Output[VAR]
        Out_m_Data02 AT %QB2563: WORD; (*  *) [CHANNEL (Q)] = 24
      WORD Output[VAR]
        Out_m_Data03 AT %QB2565: WORD; (*  *) [CHANNEL (Q)] = 0
  Modbus TCP Master[FIX]
DNP3 Protocol[FIX]
```

Рис. 8. Окно «Конфигурация ПЛК»

Имя	Тип	Адрес (1 .. 65536)	Обработка	Значение	Описание
Out1	Input Register	1		42	
Out2	Input Register	2		0	

Рис. 9. Окно «Конфигурация OPC-сервера»

Out_m_Data01 – выходная переменная Modbus.

Считаем её адрес. Затем к этому адресу прибавляем 2 (2 байта) и получаем адрес второй выходной переменной *Modbus*.

В самой среде CODESYS показано, что значения предаются на выход (рис. 8):

Но в OPC-сервере передаётся значение только для первой переменной (рис. 9):

В чём может быть проблема?

Например, задача состоит в том, что существует некий внутренний массив данных *Data_internal[i]*, и пользователь хочет из этого массива передавать данные на выходные переменные *Modbus*.

Если там большое количество переменных Modbus, то вручную это не очень удобно делать.

Ответ

Использование этого приема:

```
pt_write[1]:=ADR(Out_m_Data01);
(* и т.д. *)
```

без специальных мер приводит к тому, что среда разработки CODESYS сгенерирует единственную ссылку на выходную часть образа процесса, соответствующую *Out_m_Data01*, а остальная область *Input Registers* не будет обновляться из приложения.

В результате изменения по Modbus будут видны только для первого регистра.

«Специальные меры» состоят в том, чтобы явно сослаться на желаемую область адресов в образе процесса. Скажем, если есть такое объявление:

```
VAR CONSTANT
        ARRAY_SZ : INT := 3;
END_VAR
VAR
        Area:ARRAY[1..ARRAY_SZ]  OF
WORD:=5,6,7;
MbSlaveArea AT %QB2561 : ARRAY
[1..ARRAY_SZ] OF WORD;
        p_mb : POINTER TO WORD;
        idx : INT;
END_VAR,
```

то в теле программы достаточно написать:

```
MbSlaveArea;
```

а затем работать с указателями по следующему примеру:

```
p_mb := ADR(MbSlaveArea[1]);
FOR idx := 1 TO ARRAY_SZ DO
        p_mb^ := Area[idx];
        p_mb := p_mb +
SIZEOF(p_mb^);
END_FOR
```

Результат выполнения программного кода представлен на рис.10.

На рис. 11 показано окно программы “Fastwel Modbus OPC Server”, где отображаются значения трёх переменных.

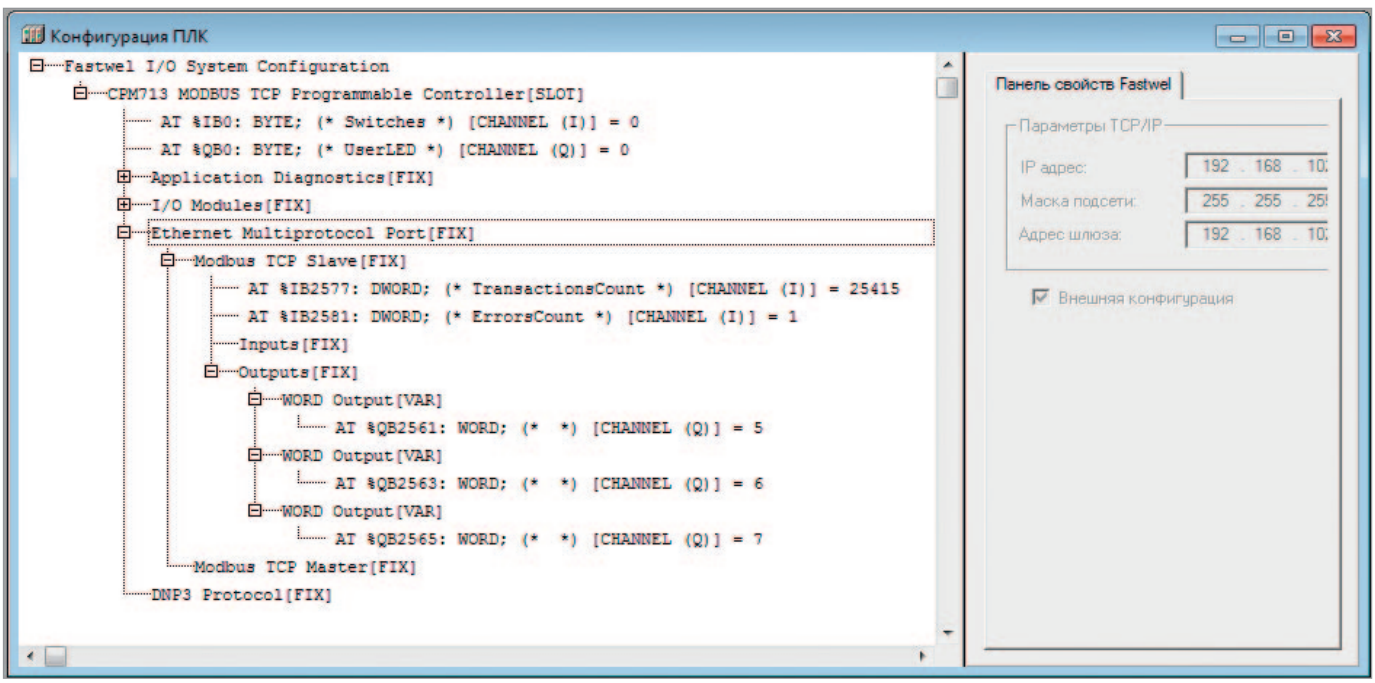


Рис. 10. Результат выполнения программного кода

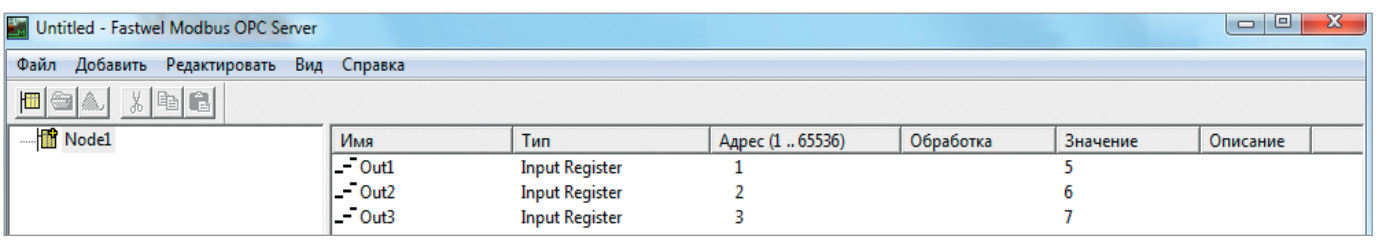


Рис. 11. Окно программы «Fastwel Modbus OPC Server»

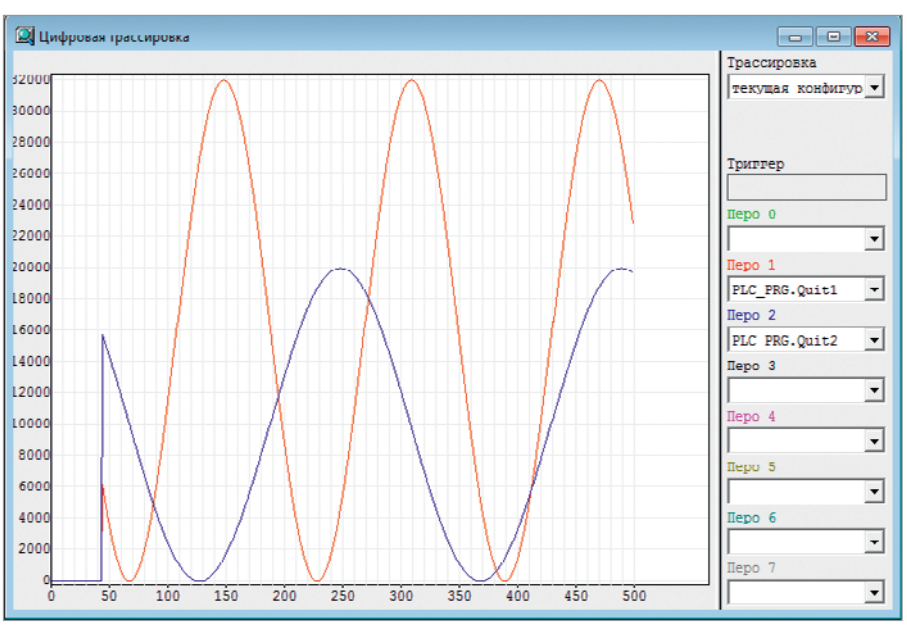


Рис. 12. Окно цифровой трассировки

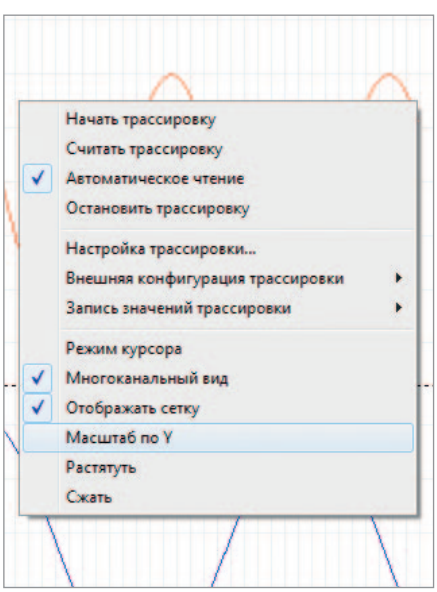


Рис. 13. Контекстно-зависимое меню в окне цифровой трассировки

Fastwel

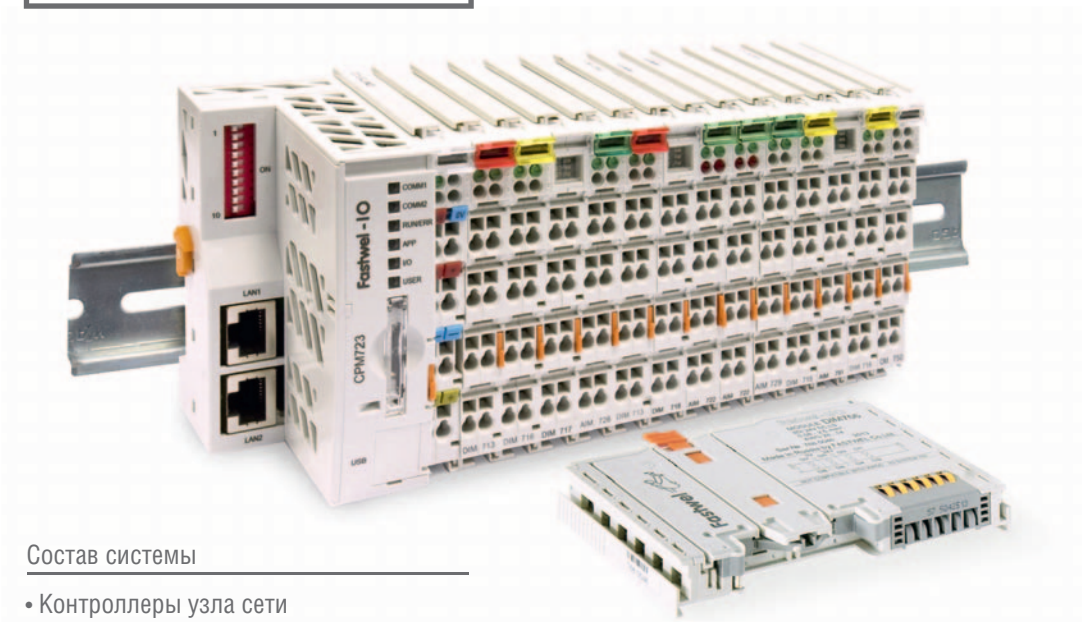
Российская электроника для ответственных применений

Распределённая система ввода-вывода FASTWEL I/O

МОРСКОЙ РЕГИСТР
ПОЖАРНЫЙ СЕРТИФИКАТ
СЕРТИФИКАТ СООТВЕТСТВИЯ
РЕЕСТР СРЕДСТВ ИЗМЕРЕНИЙ

-40...+85°C

95%



Состав системы

- Контроллеры узла сети
- Модули:
  - дискретного ввода-вывода
  - аналогового ввода-вывода
  - измерения температуры
  - сетевых интерфейсов

Модульный программируемый контроллер

- Процессоры 500/600 МГц
- Встроенный и внешний флэш-накопители объёмом до 32 Гбайт
- Энергонезависимая память 128 кбайт с линейным доступом
- Бесплатная адаптированная среда разработки приложений CODESYS
- Часы реального времени
- Сервис точного времени на базе GPS/GLONASS PPS
- Модули ввода-вывода с контролем целостности цепей

CPM711
- Протокол передачи данных CANopen
- Сетевой интерфейс CAN

CPM712
- Протокол передачи данных Modbus RTU, DNP3
- Сетевой интерфейс RS-485

CPM713
- Протокол передачи данных Modbus TCP, DNP3
- Сетевой интерфейс Ethernet

CPM723
- Протоколы передачи данных Modbus TCP/RTU
- Сетевой интерфейс 2×Ethernet

www.fastwel.ru

PROSOFT®
WWW.PROSOFT.RU

ОФИЦИАЛЬНЫЙ ДИСТРИБЬЮТОР

МОСКВА
(495) 234-0636
info@prosoft.ru

САНКТ-ПЕТЕРБУРГ
(812) 448-0444
info@spb.prosoft.ru

ЕКАТЕРИНБУРГ
(912) 620-8050
ekaterinburg@regionprof.ru

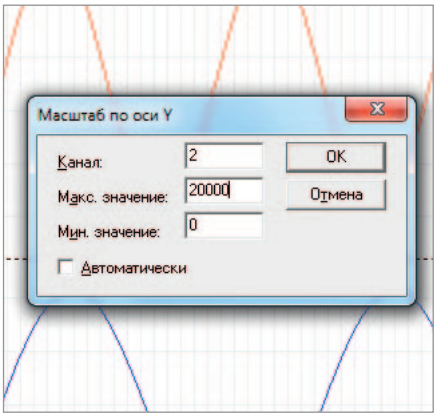


Рис. 14. Окно «Масштаб по оси *Y*»

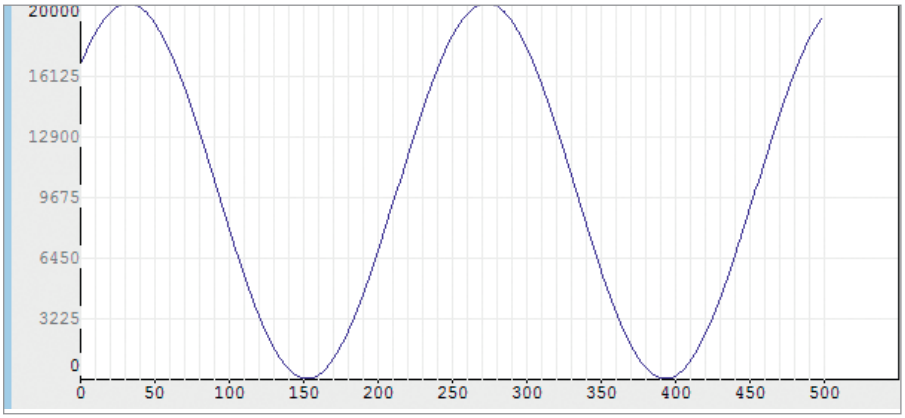


Рис. 15. Отображение трассы

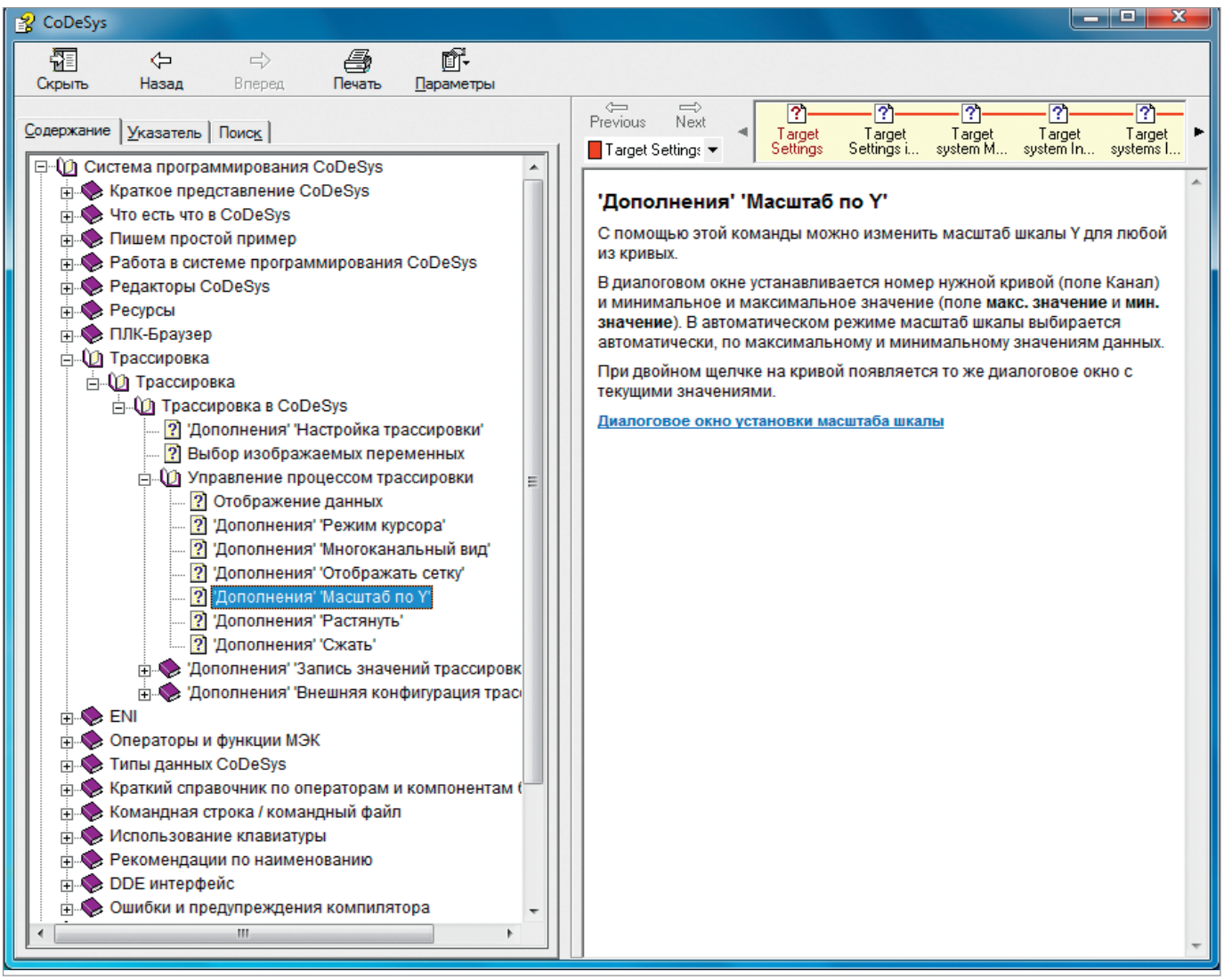


Рис. 16. Окно справочной системы CODESYS 2.3

Вопрос

При разработке модели «Синусоида» выводятся графики в окне цифровой трассировки (рис. 12). В режиме с отключённым многоканальным видом не получается задать фиксированный диапазон по оси Y. Как можно этого добиться? Работа ведётся в эмуляторе.

Ответ

Для задания фиксированного диапазона необходимо:

1. В процессе трассировки щёлкнуть правой кнопкой над областью трассировки и выбрать Масштаб по Y (рис. 13).
2. В диалоговой панели Масштаб по оси Y снять флажок Автоматически (рис. 14). Затем ввести номер трассы в поле Канал, требуемые минимальное и максимальное значения в соответствующие поля и нажать OK. Область отображения трассы для соответствующего канала будет визуализирована в заданных пределах (рис. 15).

Более подробная информация приведена в справочной системе и документации на CODESYS 2.3 (рис. 16). ●

Авторы – сотрудники компании ДОЛОМАНТ и фирмы ПРОСОФТ
Телефон: (495) 234-0636
E-mail: info@prosoft.ru

Fastwel

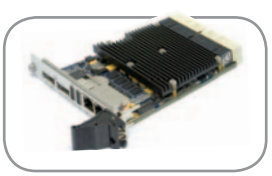
Российская электроника для ответственных применений

Скорость и надежность современных технологий

Поддерживаемые ОС

CompactPCI 2.0, 2.16, 2.30, Serial

PICMG

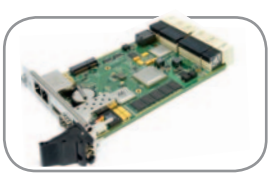


CPC512

Intel Core i7
1×Gbe, 2×PCIe x8, 4×PCIe x4 для межмодульной коммутации

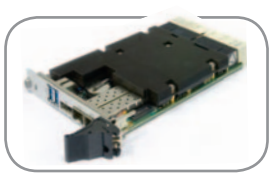
CPC514

Эльбрус-4С
8 Gb RAM, 16 Gb SSD, 3×SATA II, 9×USB 2.0, 3×Gbe



CPC516

Байкал-Т
5×PCIe 1.0, SATA III, 2xGbe, DP 1920×1080@60 fps



CPC518

Intel Xeon D
32 Gb DDR4, 24×PCIe 3.0, 2×SPF + 10Gbe, DP 1920×1440@60 fps

PROSOFT®
WWW.PROSOFT.RU

ОФИЦИАЛЬНЫЙ ДИСТРИБЬЮТОР

МОСКВА
(495) 234-0636
info@prosoft.ru

САНКТ-ПЕТЕРБУРГ
(812) 448-0444
info@spb.prosoft.ru

ЕКАТЕРИНБУРГ
(343) 356-5111
info@prosoftsystems.ru
(912) 620-8050
ekaterinburg@regionprof.ru

PROSOFT® SYSTEMS

РОССИЙСКИЕ КОНТРОЛЛЕРЫ REGUL

для автоматизации технологических процессов в газовой промышленности

Высокопроизводительные решения для создания АСУ ТП любой сложности, а также локальных и распределенных систем управления на объектах газового хозяйства

ПЛК REGUL R600

ПЛК REGUL R500

ПЛК REGUL R400

ПЛК REGUL R200

- Полностью российский продукт с оперативной техподдержкой
- «Бесшовная» интеграция линейки контроллеров на основе единой шины обмена данными
- Единая бесплатная среда разработки прикладного ПО Epsilon LD с поддержкой 5 языков стандарта IEC 61131-3
- «Горячая» замена модулей
- Поддержка различных схем резервирования
- Время цикла от 1 мс
- Синхронизация времени по GPS/GLONASS с точностью 50 мкс
- Встроенные архивы
- Поддержка визуализации
- Работа в экстремальных условиях (расширенный температурный диапазон, повышенная влажность и агрессивные среды)

Контроллеры серии REGUL RX00 применимы в широком спектре задач автоматизации газового комплекса. На их основе реализованы сотни ответственных внедрений на территории России и стран СНГ

Подробная информация о контроллерах и примеры проектов – на сайте **prosoftsystems.ru**

Инженерная компания ООО «Прософт-Системы»
620102, г. Екатеринбург, ул. Волгоградская, 194а
тел.: +7 (343) 3-565-111
prosoftsystems.ru