

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 СРМ713?

ВОПРОСЫ-ОТВЕТЫ

Ответ

Оценить уровень загрузки процессора **СРМ713** 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 $F_lecTasks_getInfo$ 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: <u>TYPE F_TASK_INF0</u> :

STRUCT

period_us : DWORD;

(* период циклической задачи в мкс или для ациклической задачи — 16#FFFFFFF *)



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



Рис. 3. Пример использования функции F_IecTasks_getInfo

🍤 PLC	C_PRG (PRG-ST)
0003	⊟Task_info
0004	period_us = 80000
0005	.cyclesCount = 695
0006	.overrunsCount = 0
0007	.minExecutionTime_us = 40
8000	.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 lecTasks 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_lecTasks_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_lecTasks_getInfo* приведён на рис. 3 и 4.

Для рассмотренной в примере задачи: period us = 80000 — периол 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 СРМ713 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 СОМ-порт с помощью кабеля, входящего в комплект поставки.

В случае если контроллер СРМ713 уже использовался ранее, но по какимто 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 СРМ713 или его МАС-адрес с целью привязать программу к данному контроллеру?

Ответ

Установить, а затем прочитать пользовательский серийный номер контроллера СРМ713 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 СРМ713 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];

CoDeSys - Установка	_Чтение серииного номера.pro - [PLC_PRG (PRG-ST)]
🎭 <u>Ф</u> айл <u>П</u> равка П	<u>роект Вставить Дополнения Онлайн Окно С</u> правка
POU I PLC_PRG (PRG)	0001 PROGRAM PLC_PRG 0002 VAR 0003 (*Переменная для записи результата вызова функций библиотеки FastwelPlatformControl.lib*) 0004 plResult:F_PLCTL_RESULT; 0005 (*Shaчeние серийного номера контроллера*) 0006 plSerialNumber:DWORD; 0008 ErrorCount:DWORD;=0; 0009 END_VAR 0010 0011 VAR CONSTANT 0012 0013 msg_debug:STRING:='Debug Info'; 0014 END_VAR
	0001 (*Установка серийного номера*) 0002 plResult:=FwPlatformSetSerialNumber(1823); 0003 (*Проверяем результат операции*) 0004 IF plResult<>F_PLCTL_OK THEN 0005 ErrorCount:=ErrorCount+1; 0006 END_IF 0007 (*Чтение серийного номера*) 0008 plResult:=FwPlatformGetSerialNumber(ADR(plSerialNumber)); 0009 IF plResult<>F_PLCTL_OK THEN 0010 ErrorCount:=ErrorCount+1; 0011 FnlResult 0012

Рис. 6. Пример использования функций библиотеки FastwelPlatformControl.lib

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		
	0001 plResult = F_PLCTL_OK	
	0002 plSerialNumber = 1823	
······≣] PLC_PRG (PRG)	0003 ErrorCount = 0	
	0004 SERIAL_NUMBER = 1823	
	0005 msg_debug = 'Debug Info'	
	0006	
	0007	
	8000	
	0001 (*Установка серийного номера*)	
	0002 plResult:=FwPlatformSetSerialNumber(1823);	plResult = F PLCTL OK
	0003 (*Проверяем результат операции*)	
	0004 IF plResult<>F_PLCTL_OK THEN	plResult = F_PLCTL_OK
	0005 ErrorCount:=ErrorCount+1;	ErrorCount = 0
	0006 END_IF	
	0007 (*Чтение серийного номера*)	
	0008 plResult:=FwPlatformGetSerialNumber(ADR(plSerialNumber));	plResult = F_PLCTL_OK plSerialNumber = 1823
	0009 IF plResult<>F_PLCTL_OK THEN	plResult = F_PLCTL_OK
	0010 ErrorCount:=ErrorCount+1;	ErrorCount = 0
	0011 END_IF	

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 ОРС-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_DataO1); (* 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;

a 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

🛅 Uni	titled - Fastw	el Modbus OPC Se	erver							X
Файл	Добавить	Редактировать	Вид	Справка						
	3	e C								
····	Node1			Имя	Тип	Адрес (1 65536)	Обработка	Значение	Описание	
				- Out1	Input Register	1		5		
			-	Out2	Input Register	2		6		
			- I-	- Out3	Input Register	3		7		





-		-
	Считать грассировку	
	Остановить трассировку	
	Настройка трассировки	
	Внешняя конфигурация трассировки	
	Запись значений трассировки	
	Режим курсора	
1	Многоканальный вид	
1	Отображать сетку	
	Масштаб по Ү	
	Растятуть	
	Сжать	

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

Рис. 12. Окно цифровой трассировки



Рис. 14. Окно «Масштаб по оси У»







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 Ү. Как можно этого добиться? Работа ведётся в эмуляторе.

Ответ

Для задания фиксированного диапазона необходимо:

- 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 ОК. 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