2017

СПК

Настройка обмена по протоколу CAN

Руководство для начинающих пользователей

Версия: 1.1
Дата: 15.08.2017

Оглавление

1. Цель документа

Этот документ представляет собой руководство по настройке обмена данными с использованием протокола **CAN** для панельных контроллеров Овен СПК2xx.04 в среде **CODESYS V3.5**. Подразумевается, что читатель обладает базовыми навыками работы с **CODESYS** и **СПК**, поэтому общие вопросы (например, создание и загрузка проектов) в данном документе не рассматриваются; они подробно описаны в документах **СПК. Первый старт** и **СПК. FAQ**, которые доступны на сайте Овен в разделе **CODESYS V3/Документация**.

Обратите внимание, что порт **CAN** присутствует только на СПК модификаций **04** (например, СПК207.**04**, СПК207.**04**-WEB)

ВНИМАНИЕ! Модификация СПК207.04 снята с производства в конце 2016-го года.

Документ содержит пример настройки обмена с модулем **Marathon IOremote** по протоколу **CANopen**.

2. Основные сведения о работе с CAN на СПК

CAN (англ. *Controller Area Network* — сеть контроллеров) — стандарт промышленной сети, ориентированный на объединение в единую сеть различных исполнительных устройств и датчиков. Режим передачи — последовательный, широковещательный, пакетный.

CAN разработан компанией Robert Bosch GmbH в середине 1980-х и в настоящее время широко распространён в автомобильной автоматике, промышленной автоматизации, технологиях «умного дома» и других областях.

Контроллеры **СПК2xx.04(-WEB)** поддерживают работу в сети **CAN** по протоколу верхнего уровня CANopen в режиме **Master**. Диапазон возможной скорости обмена: 10 бит/с – 1 Мбит/с.

Необходимым условием для организации обмена является наличие терминирующих резисторов (120 Ом) на обоих концах линии связи. Порт СПК не имеет встроенного терминирующего резистора.

Обратите внимание, что **СПК2xx.04(-WEB)** не поддерживают другие протоколы верхнего уровня (**DeviceNet**, **J1939** и т.д.) и работу в режиме **Slave**.

Для настройки обмена со slave-устройством требуется **.eds** файл, который устанавливается в **CODESYS 3.5** и содержит описание устройства с картой регистров.

Подробная информация о стандарте и его реализациях доступна в сети Интернет – например, на сайте http://can.marathon.ru/.

3. Пример: СПК207.04 + Marathon IOremote

Рассмотрим пример настройки обмена по протоколу **CANopen**. В нем мы наладим связь между контроллером **СПК207.04** (CAN master) и модулем дискретного ввода-вывода **Marathon IOremote R2DIO 8/8** (CAN slave).

Рис. 3.1. Внешний вид модуля **Marathon IOremote R2DIO 8/8**

Модуль сконфигурирован на следующие настройки:

Табл. 3.1. Настройки модуля **Marathon IOremote R2DIO 8/8**

Параметр	**Значение**
Протокол	CANopen
Номер CAN узла (Node ID)	2
Скорость CAN сети	125 кбит/с

Пример создан в среде **CODESYS 3.5 SP7 Patch4** и подразумевает запуск на **СПК207.04.CS(-WEB)** с таргет-файлом **3.5.2.40 (023).**

Пример доступен для скачивания: Example_CAN.projectarchive

1. В первую очередь необходимо установить в **CODESYS** файл описания slave-устройства (**.eds** файл). Обычно он доступен на диске с ПО из комплекта поставки.

В **CODESYS** выберите вкладку **Инструменты** и запустите **Репозиторий устройств**. Нажмите кнопку **Установить** и укажите путь к файлу **.eds** (не забудьте в диалоговом окне указать соответствующий формат файла).

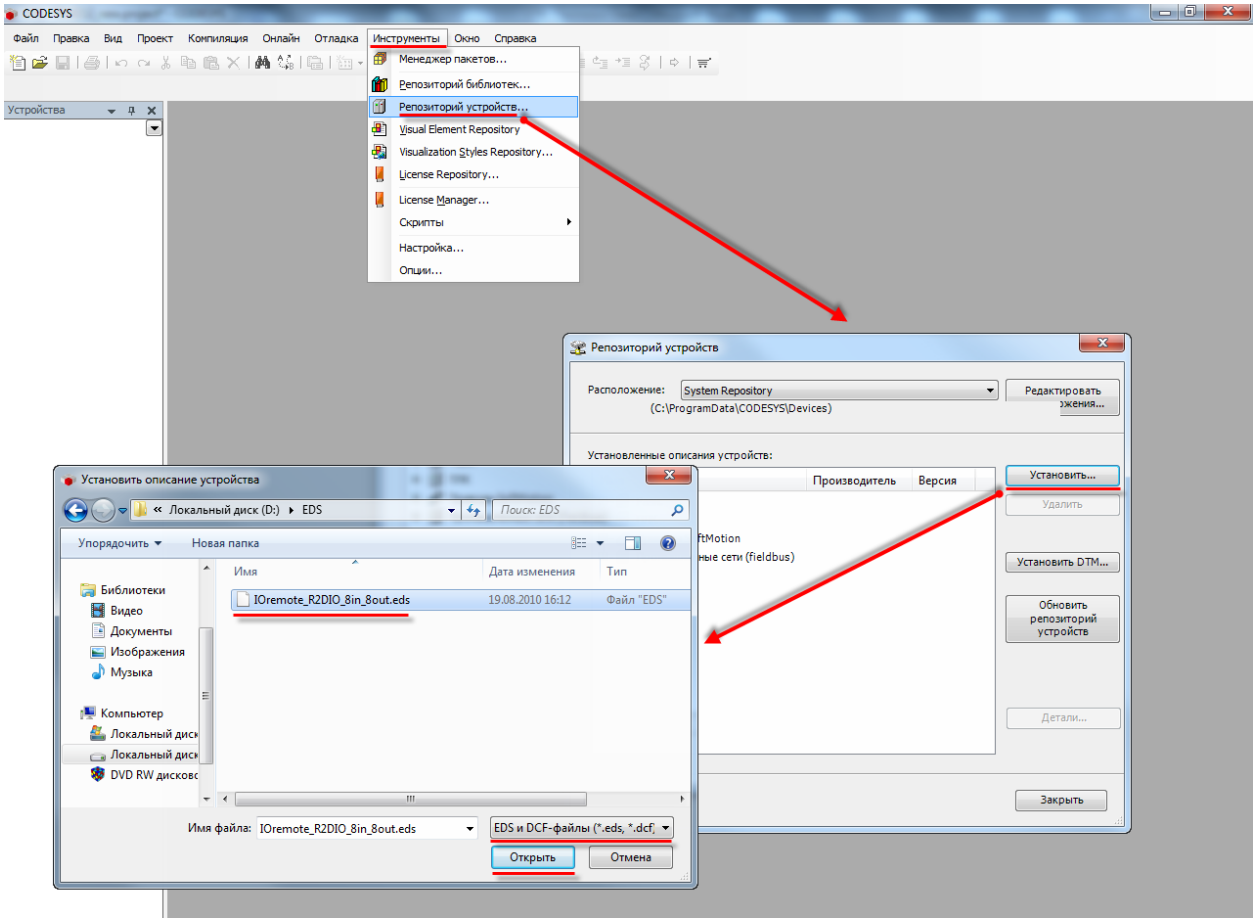

Рис. 3.2. Установка **.eds** файла в **CODESYS**

2. Создайте новый проект для контроллера **СПК207.04.CS(-WEB)**. Язык программы не имеет значения.

3. В программе **PLC_PRG** объявите следующие переменные:

```
PLC_PRG
1  PROGRAM PLC_PRG
2  VAR
3      byInputs    :BYTE;      // входы модуля в виде битовой маски
4      byOutputs   :BYTE;      // выходы модуля в виде битовой маски
5  END_VAR
```

Рис. 3.3. Объявление программы **PLC_PRG**

4. Нажмите **ПКМ** на компонент **Device** и добавьте компонент **CANbus**, расположенный во вкладке **Промышленные сети/CANbus.**

Обратите внимание, что версия компонента не должна превышать версию **target-файла** СПК. Для отображения предыдущих версий компонента поставьте галочку **Отображать все версии**.

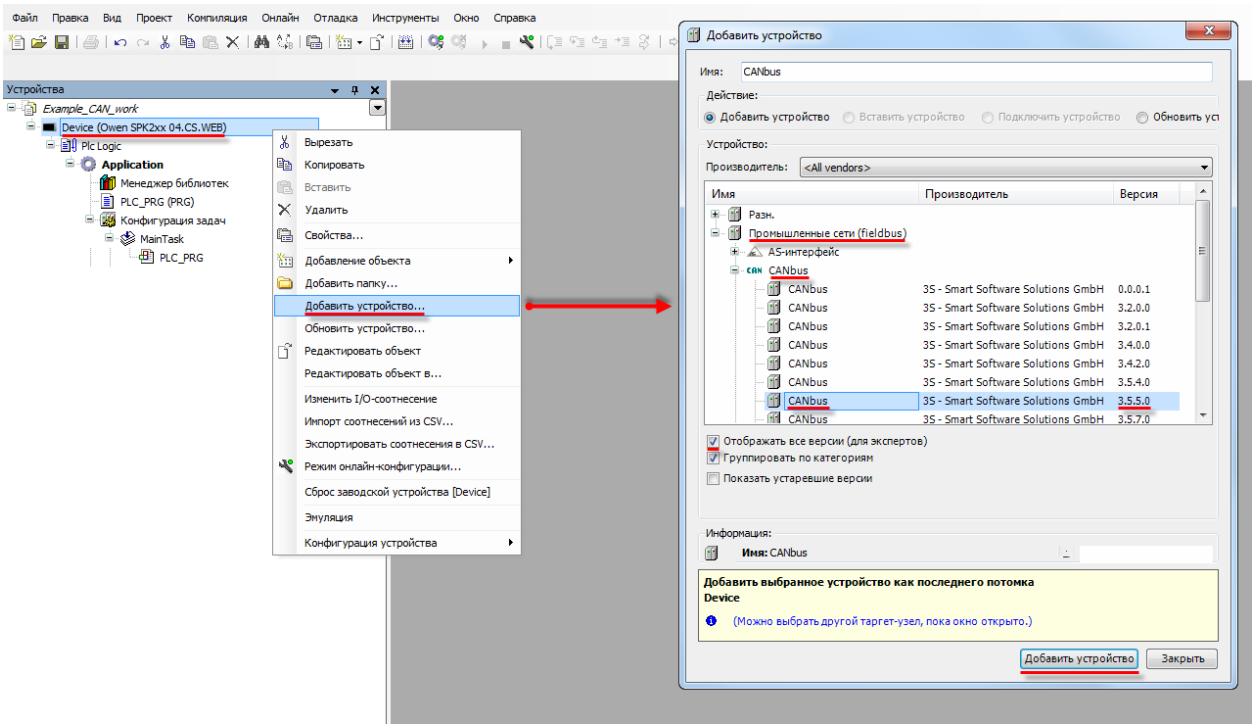

Рис. 3.4. Добавление компонента **CANbus**

В настройках компонента на вкладке **Общее** укажите номер используемой CAN-сети (у СПК только один порт **CAN** с номером сети **0**) и скорость передачи (в соответствии с табл. 3.1. **– 125000 бит/с**).

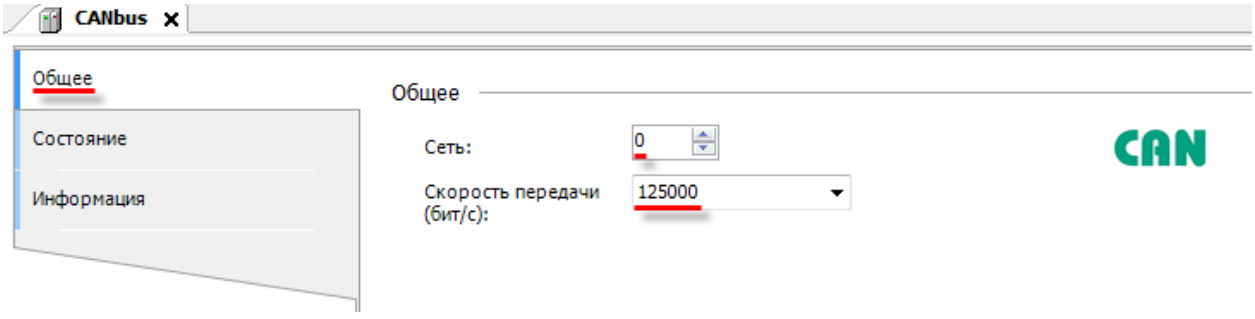

Рис. 3.5. Настройки компонента **CANbus**

5. Нажмите **ПКМ** на компоненте **CANbus** и добавьте компонент **CANopen Manager**, расположенный во вкладке **Промышленные сети/CANopen/CANopen Manager.**

Обратите внимание, что версия компонента не должна превышать версию **target-файла** СПК. Для отображения предыдущих версий компонента поставьте галочку **Отображать все версии**.

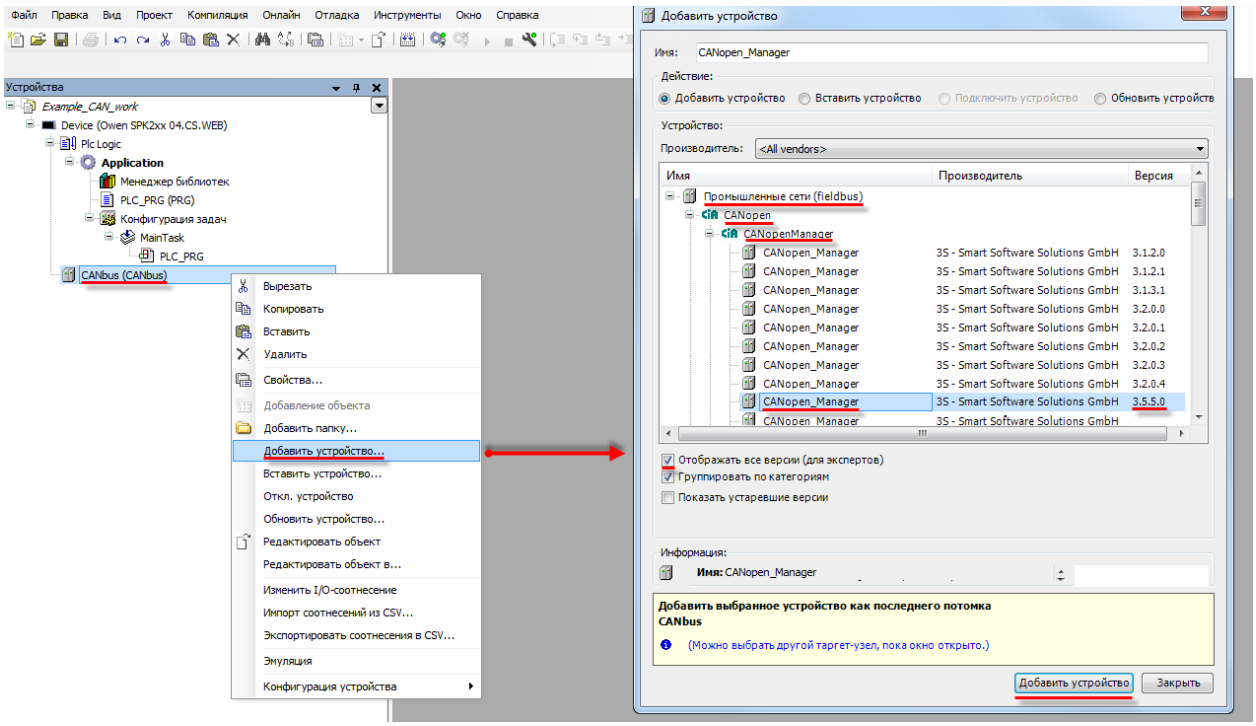

Рис. 3.6. Добавление компонента **CANopen Manager**

Оставим настройки компонента в значениях по умолчанию. Их подробное описание приведено в справке **CODESYS**.

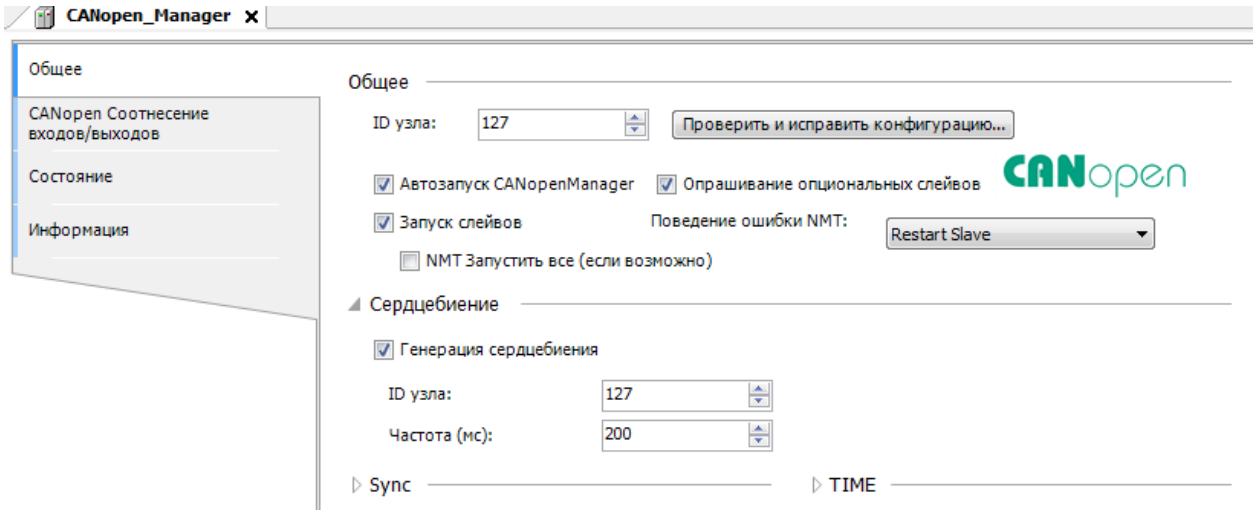

Рис. 3.7. Настройки компонента **CANopen Manager**

6. Нажмите **ПКМ** на компоненте **CANopen Manager** и добавьте нужное slave-устройство - в нашем примере таковым является модуль **Marathon IOremote R2DIO.**

Обратите внимание, что для добавления slave-устройства в проект необходимо предварительно установить его **.eds** файл в **Репозиторий устройств** (см. пп. 1).

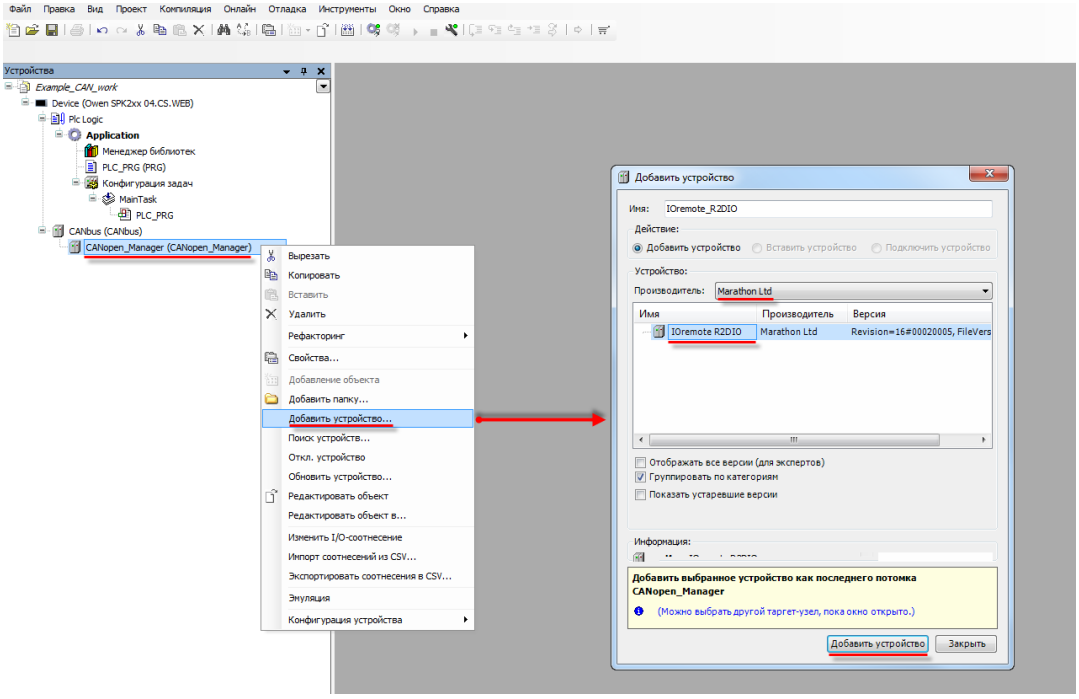

Рис. 3.7. Добавление CAN Slave устройства

В настройках компонента на вкладке **Общее** укажите ID узла (в соответствии с табл. 3.1 он равен **2**). Поставьте галочку **Экспертные установки**, чтобы получить доступ к дополнительным настройкам. Поставьте галочку **Опц. устройство** – при ее отсутствии обмен с модулем происходить не будет. Подробное описание остальных настроек приведено в справке **CODESYS**.

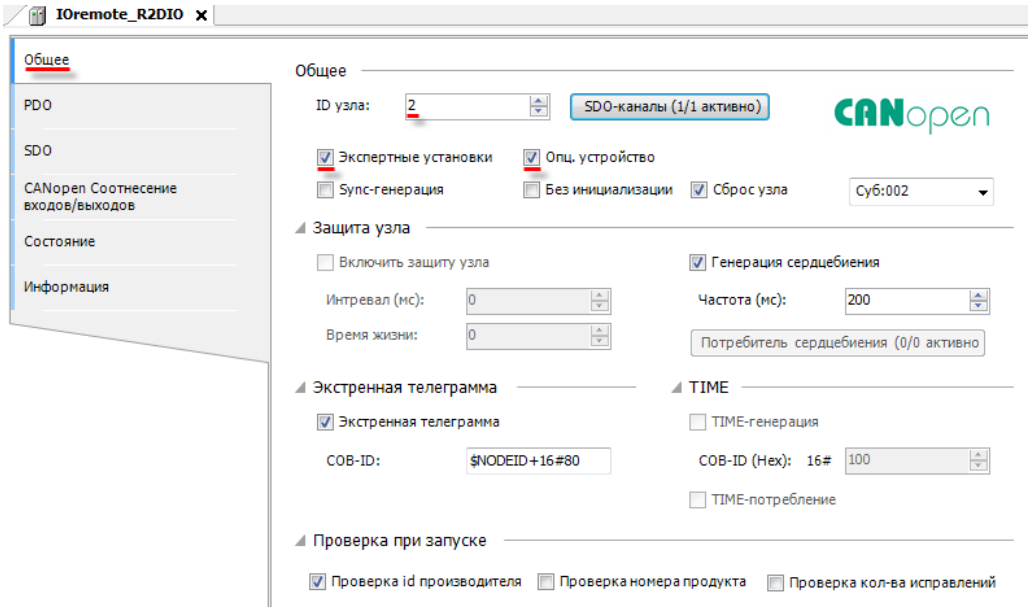

Рис. 3.8. Настройка CAN Slave устройства

На вкладке **PDO** поставьте галочки рядом с параметрами, которые будут записываться в модуль (первая таблица) и считываться из модуля (вторая таблица). У нашего модуля мы будем записывать параметр **Write Output 1h to 8h** (состояния дискретных выходов в виде битовой маски) и считывать параметр **Read Input 1h to 8h** (состояния дискретных входов в виде битовой маски).

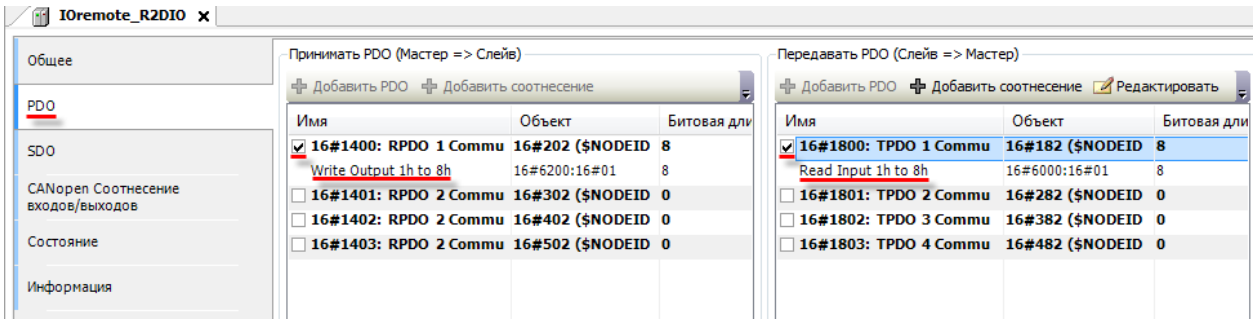

Рис. 3.9. Настройка **PDO** CAN slave устройства

На вкладке **CANopen Соотнесение входов/выходов** привяжите к **PDO** переменные программы, а для параметра **Всегда обновлять переменные** выберите значение **Вкл. 2 (Всегда в задаче цикла шины)**.

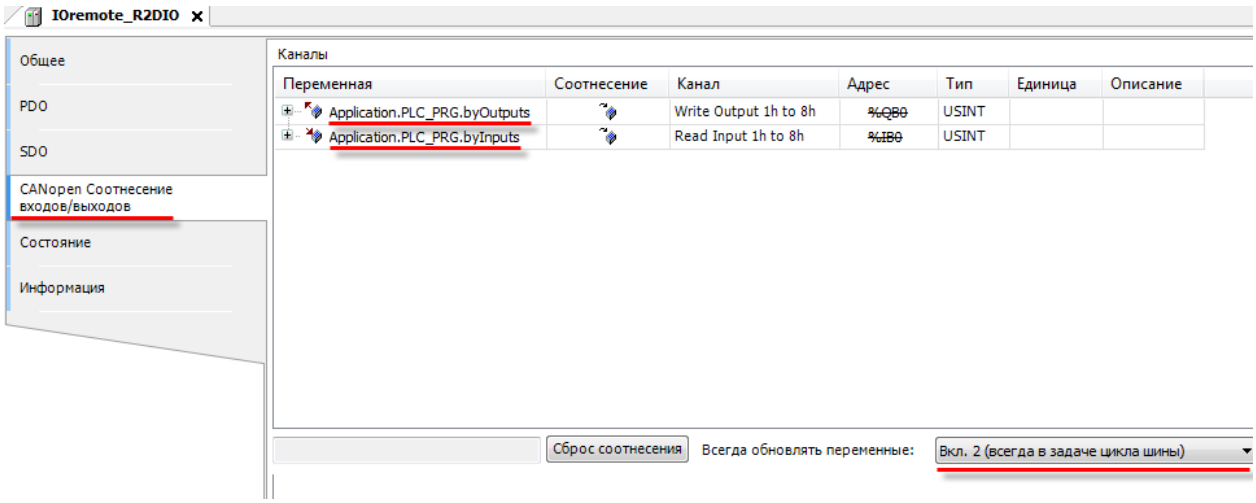

Рис. 3.10. Привязка переменных к **PDO**

7. Добавьте в проект экран визуализации. В его свойствах укажите разрешение **800x480**. Добавьте на экран 8 элементов **Индикатор** (для отображения состояний входов модуля) и 8 элементов **Выключатель** (для управления выходами модуля).

К индикаторам привяжите биты переменной byInputs: **byInputs.0**, **byInputs.1** … **byInputs.7**.

К выключателям привяжите биты переменной byOutputs: **byOutputs.0, byOutputs.1** … **byOutputs.7**.

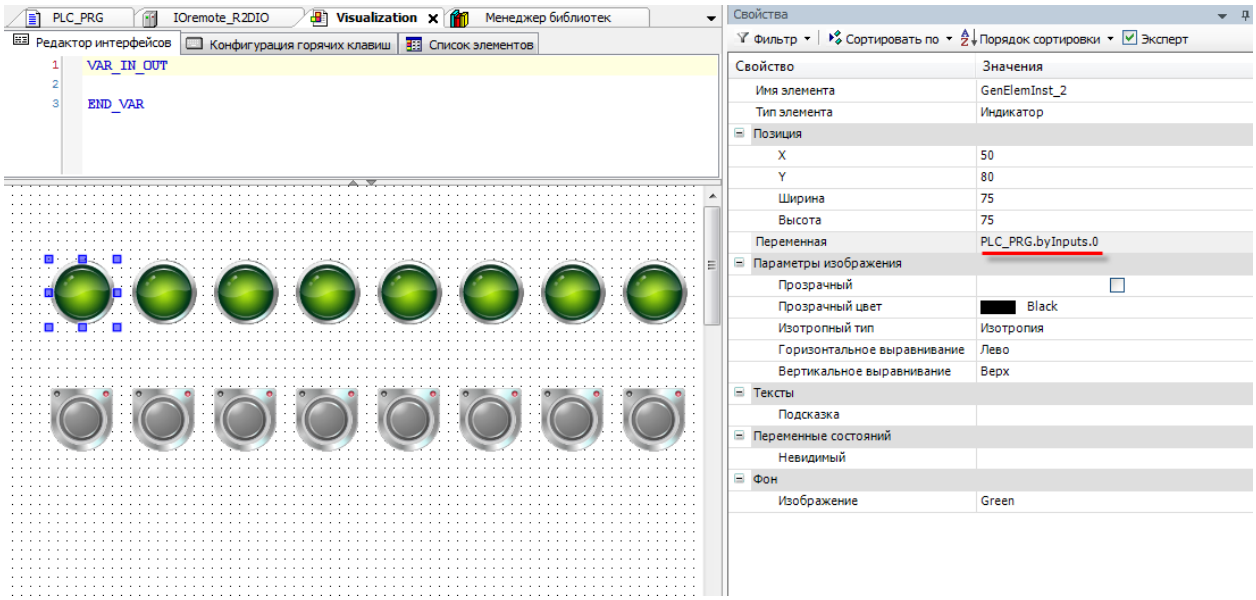

Рис. 3.11. Привязка битов переменных к элементам визуализации

Более подробно вопросы создания экранов визуализации рассмотрены в документе **СПК. Визуализация.**

8. Соедините СПК и модуль по интерфейсу CAN. Загрузите проект в СПК и запустите его. Индикаторы будут отображать состояния входов модуля. С помощью нажатия на выключатели будет производиться изменение состояний выходов модуля.

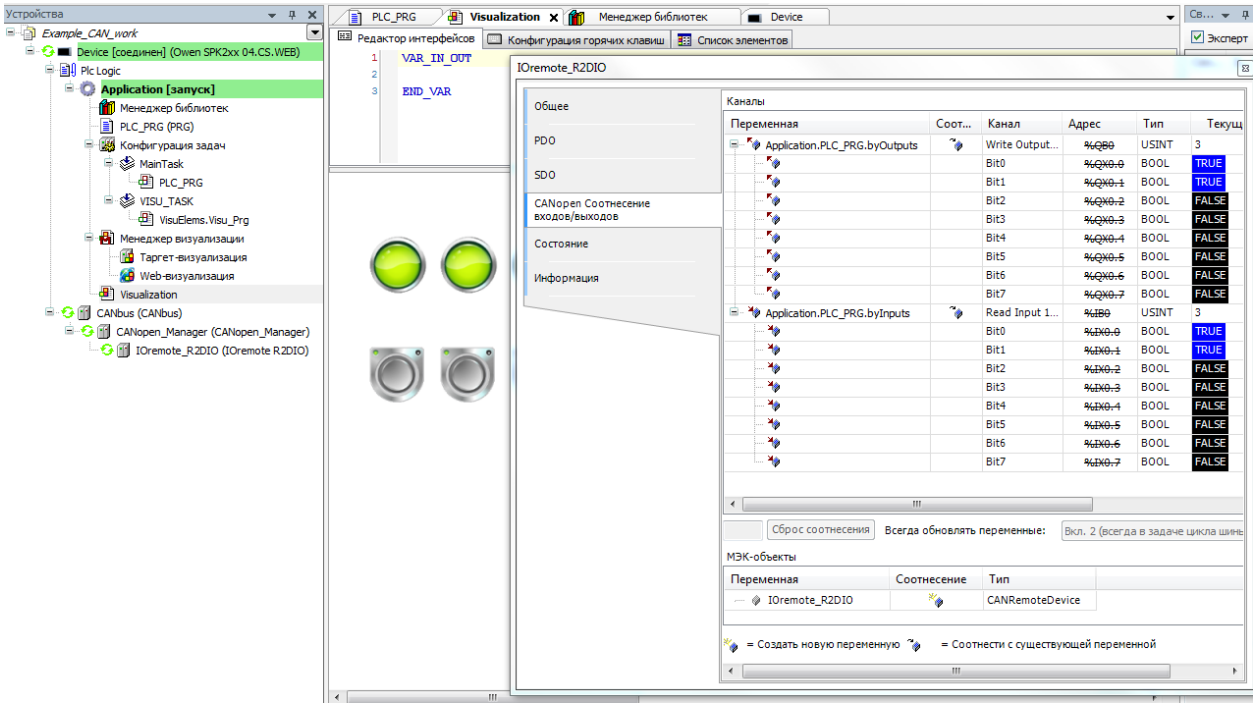

Рис. 3.12. Запуск примера на СПК