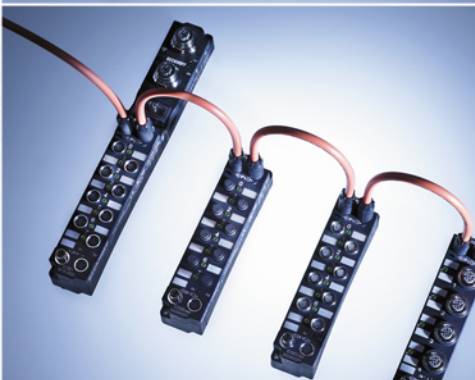
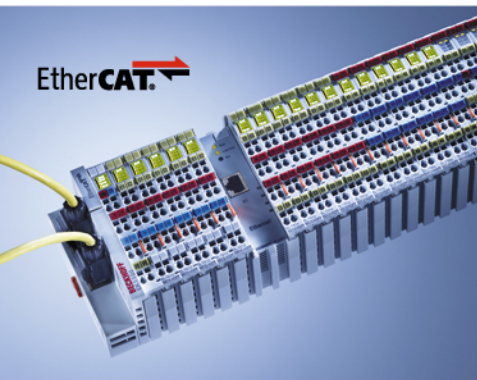
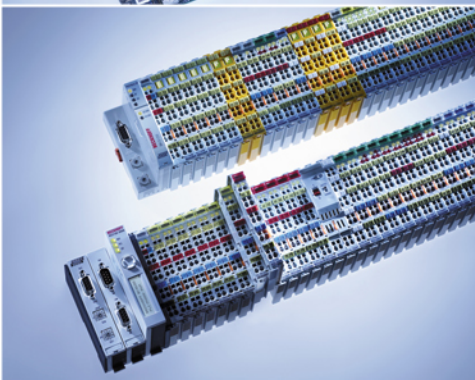


New Automation Technology

Работа с устройствами EnOcean





Пример EnOcean: Введение

Рассмотрим пример применения модуля **KL6021-0023** и беспроводного адаптера **KL6023**.

Данный пример будет реализован для контроллера CX8090 - контроллера на базе процессора ARM, соответственно может быть применен на контроллерах с процессорами ARM. (Контроллеров CX80xx, CX90xx, CP66xx, CP26xx и т.д.)

Рассмотрим библиотеки необходимые для работы с EnOcean:

Для PC систем (x86) и Embedded-PCs (CXxxxx): **TcEnOcean.lib**, **Standard.lib**, **TcBase.lib**, **TcSystem.lib**

Для контроллеров серии BCxx00: **TcEnOcean.lb6**, **Standard.lb6**, **PlcHelperBC.lb6**

Для контроллеров серий BCxx50, BCxx20 и BC9191: **TcEnOcean.lbx**, **Standard.lbx**, **TcBaseBCxx50.lbx**, **TcSystemBCxx50.lbx**

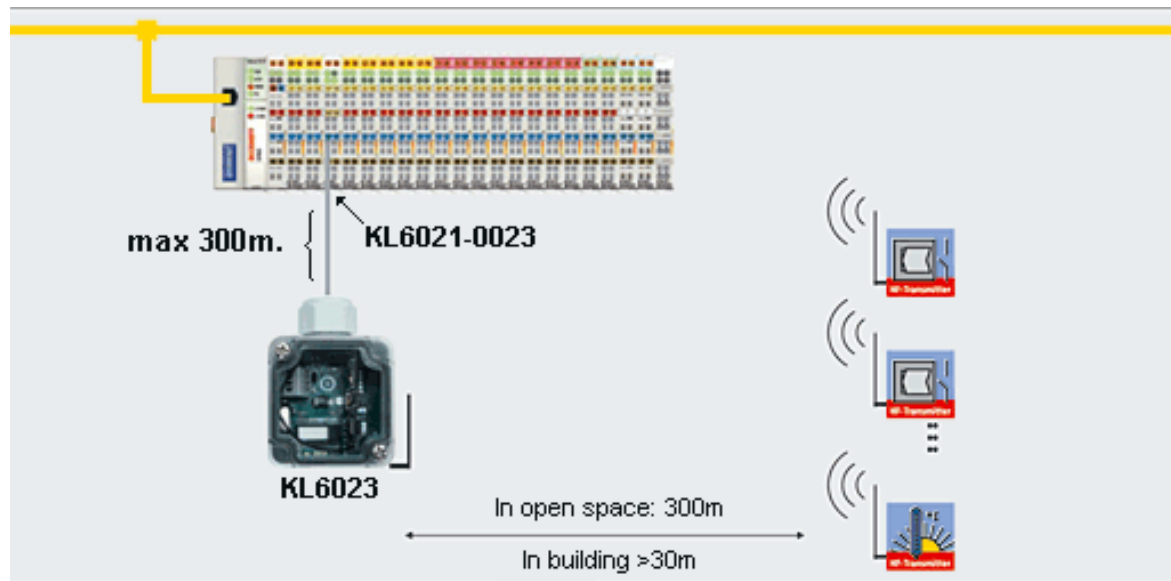
Для контроллеров серии BXxx00: **TcEnOcean.lbx**, **Standard.lbx**, **TcBaseBX.lbx**, **TcSystemBX.lbx**





Пример EnOcean: Введение

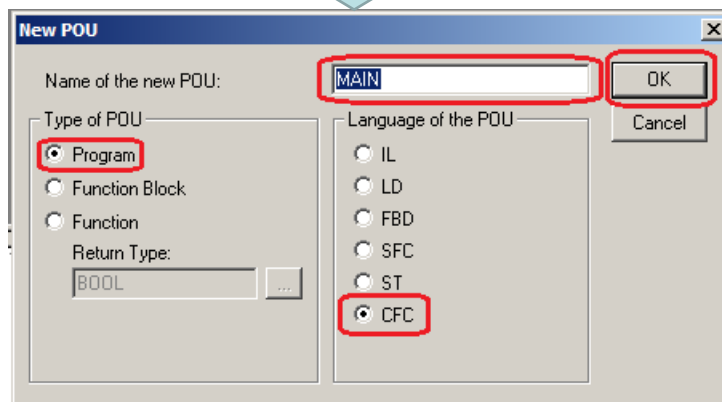
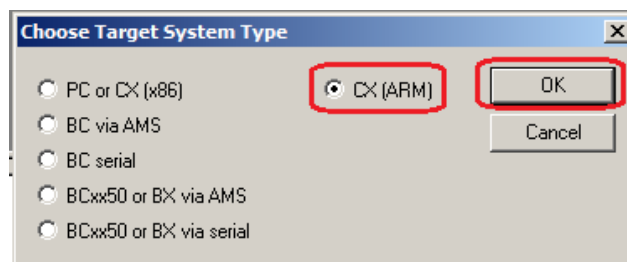
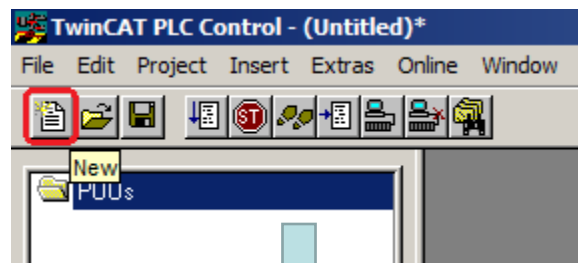
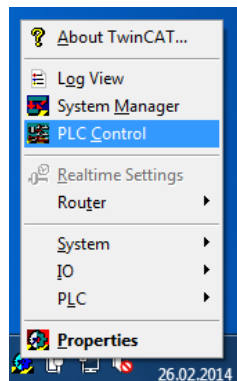
Беспроводной адаптер KL6023 получает сигналы от датчиков с EnOcean-технологией. Эти сигналы преобразуются беспроводным адаптером в сигнал RS485 и передаются далее модулем KL6021-0023



EnOcean не ограничивает количество датчиков, присвоенных одному приемнику, но рекомендуется использовать не более 100 передатчиков на один приемник, чтобы избежать одновременных передач

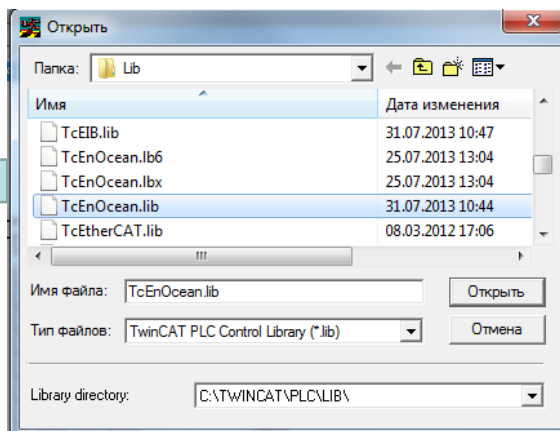
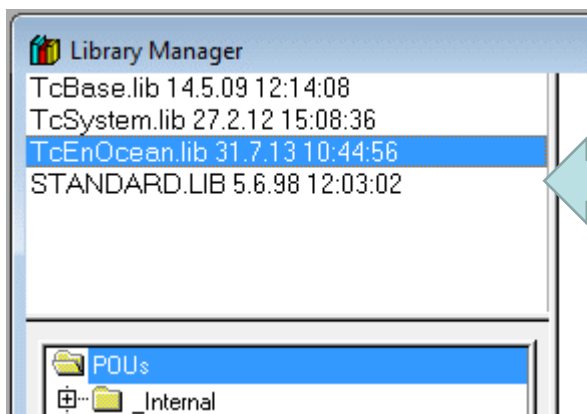
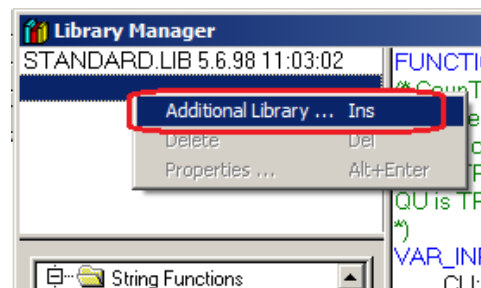
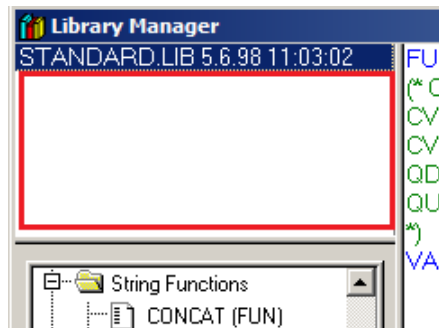
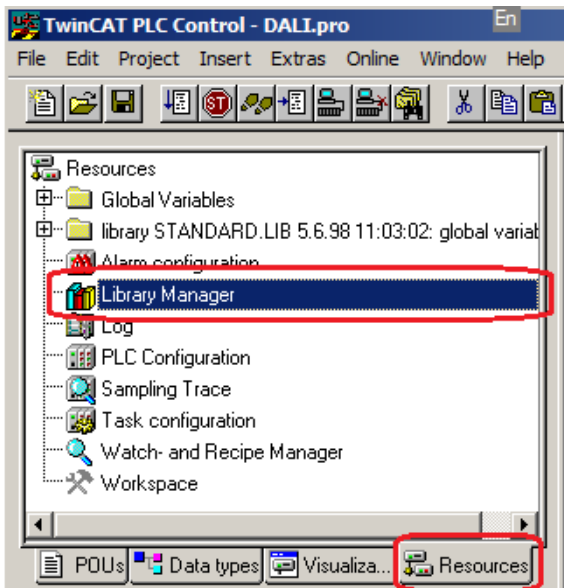


Пример EnOcean: Создание проекта



- Запускаем **TwinCAT PLC Control**
- Создаем новый проект
- Выбираем тип контроллера под который будем писать программу (В данном случае контроллер с процессором ARM)
- Указываем язык на котором будем писать программу и то что это будет именно программа, нажимаем «**OK**» (Для наглядности будем использовать язык CFC)
- Затем рекомендуем сохранить проект, для этого нажимаем на вкладку «**File**» и выбираем «**Save as...**», **путь и название проекта нужно указывать без символов кириллицы.**

Пример EnOcean: Добавление библиотек



- Во вкладке «Resources» два раза кликаем по «Library Manager»
- Видим поле с уже используемыми библиотеками (Пока лишь одной: **STANDARD.LIB**)
- Кликаем правой кнопкой мыши и выбираем пункт «Additional Library...»
- Выбираем библиотеку **TcEnOcean.lib** и нажимаем «Открыть» (Библиотеки с расширениями .lib .lbx предназначены для других типов контроллеров)
- Так как библиотека **TcEnOcean.lib** использует внутри еще ряд библиотек мы видим, что они добавились автоматически.

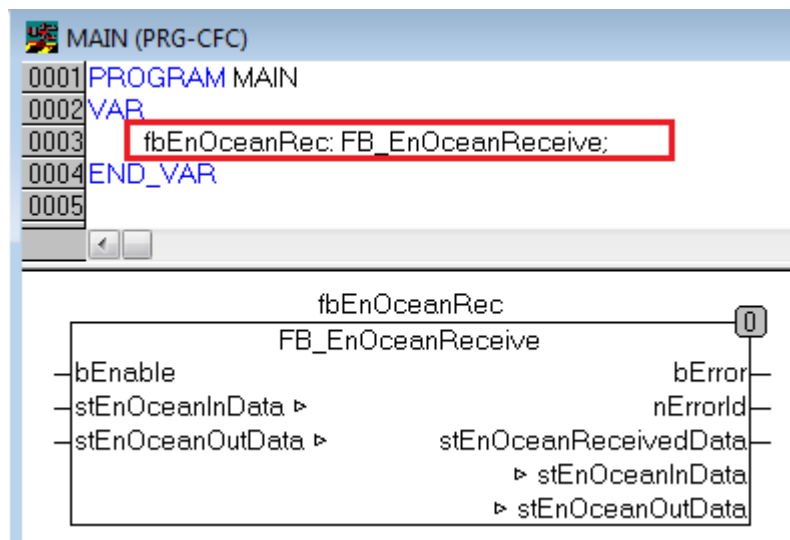
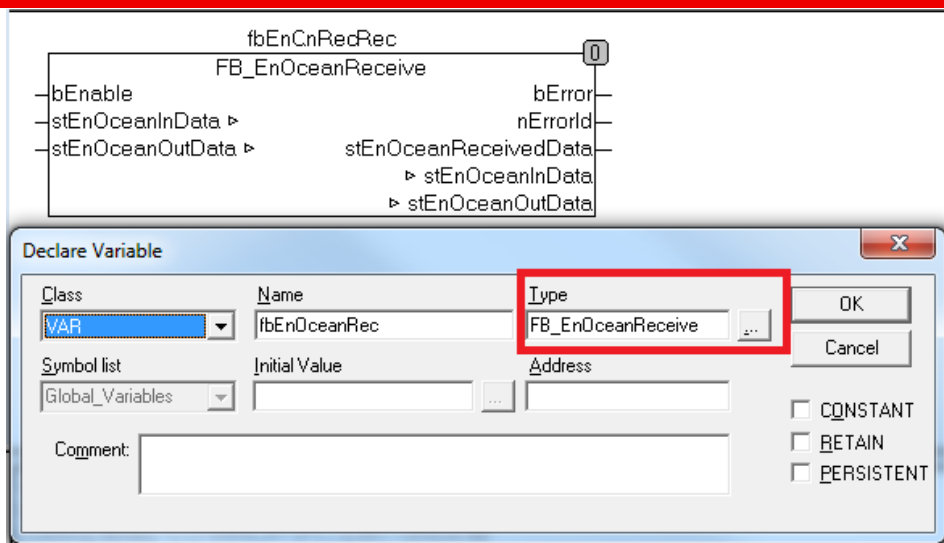
Пример EnOcean: Добавление функциональных блоков

The image illustrates the steps to add a functional block to a program in Beckhoff TwinCAT:

- Open the **POUs** window and select **MAIN (PRG)**.
- Open the **MAIN (PRG-CFC)** editor. A context menu is shown with **Box** selected.
- The selected block is added to the ladder logic, appearing as an **AND** block.
- The block is renamed to **FB_EnOceanReceive**. The diagram shows the block's inputs and outputs: **bEnable**, **stEnOceanInData**, **stEnOceanOutData**, **bError**, **nErrorId**, and **stEnOceanReceivedData** (which contains **stEnOceanInData** and **stEnOceanOutData**).

- Во вкладке «**POUs**» два раза кликаем по «**MAIN (PRG)**» в открывшемся поле кликаем правой кнопкой мыши и выбираем пункт «**Box**»
- Переименовываем получившийся тип блока в «**FB_EnOceanReceive**», блок сразу примет другой вид.
- Кликаем по «**???**» и присваиваем нашему блоку какое-то уникальное имя, после этого возникнет диалоговое окно добавления переменных в программу.

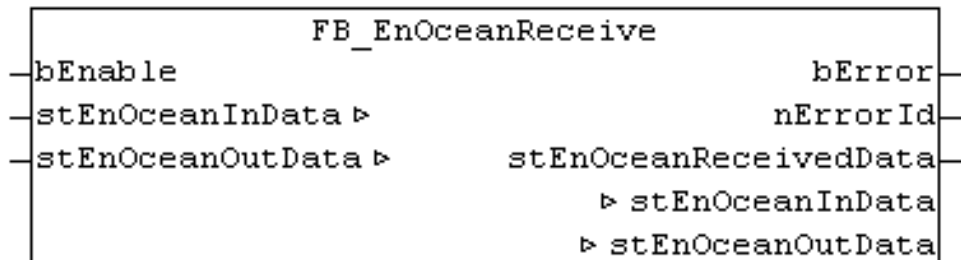
Пример EnOcean: Добавление функциональных блоков



- В окне объявления переменных обязательно нужно проверить тип переменной и если он отличается от «**FB_EnOceanReceive**» изменить его на таковой.
- Если все сделано верно то в списке переменных мы увидим новую переменную типа FB_EnOceanReceive.
- Данный блок является одним из основных при работе с протоколом EnOcean (ниже рассмотрим его подробно).

Пример EnOcean: Функциональный блок FB_EnOceanReceive

FB_EnOceanReceive



Функциональный блок FB_EnOceanReceive - собирает посылаемые передатчиками данные и затем кладет информацию в специальный структурированный тип данных

bEnable – Бит включения/выключения функции

stEnOceanInData – Входные данные, которые необходимо слинковать с модулем

KL6021-0023

stEnOceanOutData – Выходные данные, которые необходимо слинковать с модулем

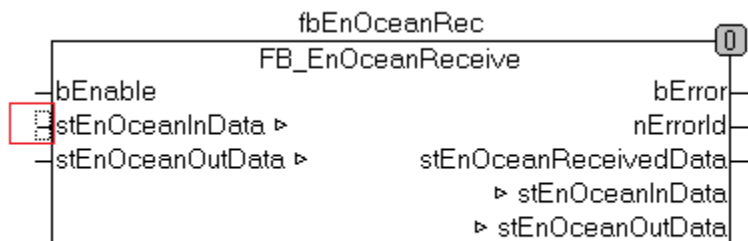
KL6021-0023

bError – Бит, свидетельствующий о возникновении ошибки в функции

nErrorId – Код возникшей ошибки

stEnOceanReceivedData – Структура в которую ф.б. кладет полученную информацию.

Пример EnOcean: Привязка данных к блоку



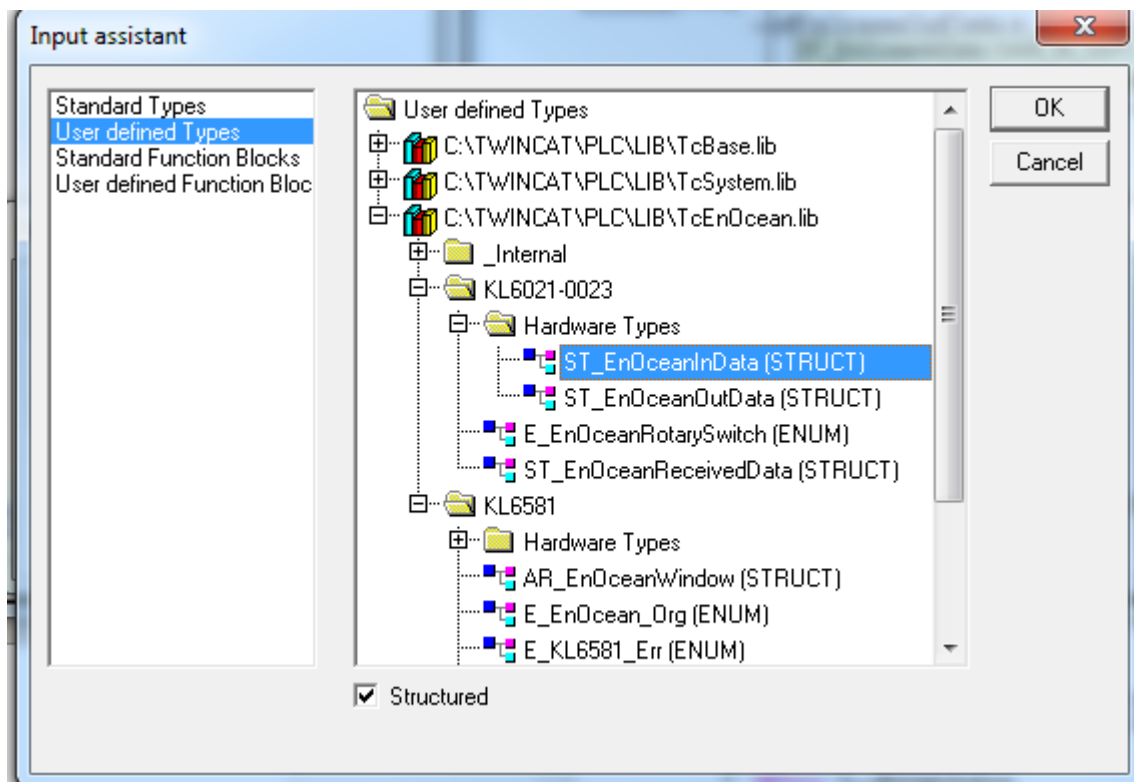
- Для того, чтобы экземпляр «**FB_EnOceanReceive**» обменивался данными с модулем KL6023, ему на вход нужно передать внешние переменные, которые впоследствии будут привязаны к аппаратной части с помощью System Manager. Для этого необходимо установить курсор интересующий вход `fbEnOceanRec` и начать вводить имя переменной.

- После ввода откроется диалоговое окно добавления переменных, в котором нужно обратить внимание, чтобы тип был «**ST_EnOceanInData**» и в поле «**Address**» указать «**%I***» - так как это входные данные

The 'Declare Variable' dialog box is shown with the following fields and options:

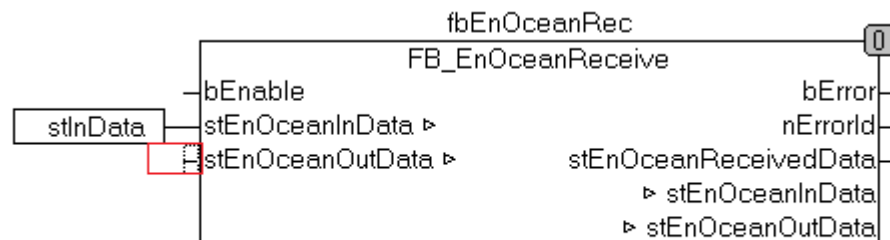
- Class:** VAR
- Name:** stInData
- Type:** ST_EnOceanInData (highlighted with a red box)
- Symbol list:** Global_Variables
- Initial Value:** (empty)
- Address:** %I* (highlighted with a red box)
- Comment:** (empty)
- Buttons:** OK, Cancel
- Options:** CONSTANT, RETAIN, PERSISTENT

Пример EnOcean: Привязка данных к блоку



- Для изменения типа данных на «**ST_EnOceanInData**» нажимаем на кнопку «...» рядом с полем «Type» и в открывшемся окне ищем интересующий нас тип данных (В данном случае он будет находиться по адресу **User defined Types - > TcEnOcean.lib -> KL6021-0023 -> Hardware Types -> ST_EnOceanInData**). Выделяем и нажимаем «**OK**».

Пример EnOcean: Привязка данных к блоку



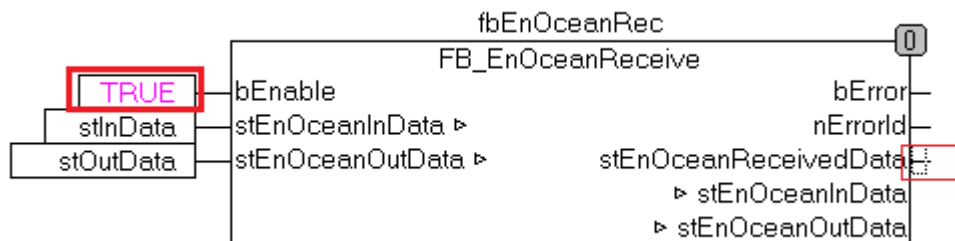
Class	Name	Type	Address
VAR	stOutData	ST_EnOceanOutData	%Q*

Comment:

CONSTANT
 RETAIN
 PERSISTENT

- Аналогичным образом добавляем данные для stEnOceanOutData.
- После ввода имени откроется диалоговое окно добавления переменных, в котором нужно на этот раз указать тип «**ST_EnOceanOutData**» и в поле «**Address**» указать «**%Q***» - так как это выходные данные.

Пример EnOcean: Привязка данных к блоку



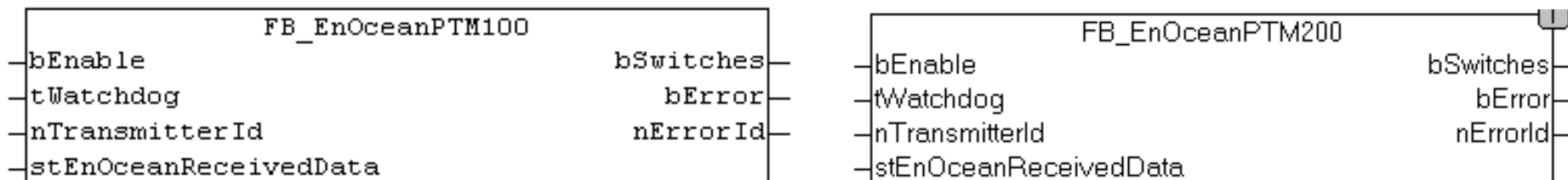
Class	Name	Type	Initial Value	Address
VAR	stResData	_EnOceanReceivedData		

- Сделаем, чтобы функциональный блок работал постоянно для этого на вход «**bEnable**» подадим значение «**TRUE**» (В случае когда указываем явное значение или переменную которая уже объявлена окно объявления переменных не открывается).
- Осталось объявить переменную типа «**ST_EnOceanReceivedData**» и записать в нее выходные данные с «**fbEnOceanRec**», делаем это абсолютно аналогично.

Пример EnOcean: Различные модули и соответствующие им функциональные блоки

- Для дальнейшего написания программы необходимо знать какие типа EnOcean модулей будут использоваться, а также желательно знать их уникальные ID. Так как для каждого типа используются свои функциональные блоки.
- Например: PTM100 модули поддерживают 8 кнопок и только 1 кнопка может быть нажата одновременно, PTM200 модули поддерживает 4 кнопки и 2 кнопки могут быть нажаты одновременно, понятно, что с данными от таких модулей нужно работать немного по разному.
- Рассмотрим подробнее функциональные блоки:
 - FB_EnOceanPTM100
 - FB_EnOceanPTM200
 - FB_EnOceanSTM100Generic
 - FB_EnOceanSTM250

Пример EnOcean: Функциональные блоки FB_EnOceanPTM100/PTM200



Функциональные блоки FB_EnOceanPTM100/PTM200 – извлекают данные из структуры «stEnOceanReceivedData» полученной ф.б. «FB_EnOceanReceive» и записывают в массив bSwitches

bEnable – Бит включения/выключения функции

tWatchdog – Время, через которое должны обновляться данные от FB_EnOceanReceive

nTransmitterId – ID передатчика, для которого назначена функция

stEnOceanReceivedData – Данные, получаемые от FB_EnOceanReceive

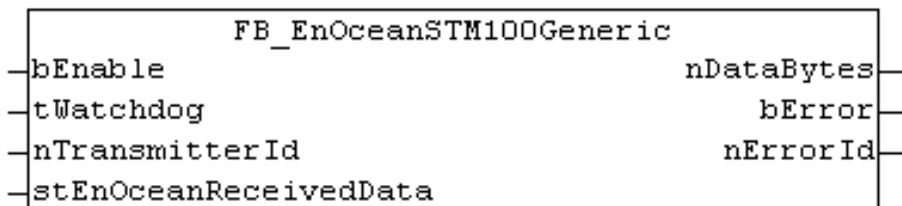
bSwitches – Состояние 8 (для PTM100) или 4 (для PTM200) кнопок на передатчике (Массив из 8 или 4 значений)

bError – Бит, свидетельствующий о возникновении ошибки в функции

nErrorId - Код возникшей ошибки

Замечание: Кнопочные выключатели передают информацию только при нажатии на кнопки, поэтому для них Watchdog можно выключить вообще ($tWatchdog = \#0s$)

Пример EnOcean: Функциональный блок FB_EnOceanSTM100Generic



Функциональные блоки

FB_EnOceanSTM100Generic – извлекают данные из структуры «stEnOceanReceivedData» полученной ф.б. «FB_EnOceanReceive» и записывает в массив nDataBytes.

bEnable – Бит включения/выключения функции

tWatchdog – Время, через которое должны обновляться данные от FB_EnOceanReceive

nTransmitterId – ID передатчика, для которого назначена функция

stEnOceanReceivedData – Данные, получаемые от FB_EnOceanReceive

nDataBytes – Преобразованный массив данные с передатчика (температура, уставка...)

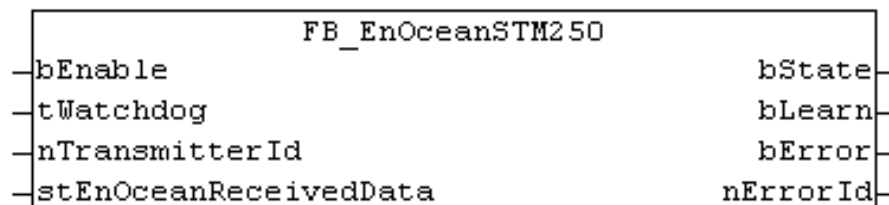
bError – Бит, свидетельствующий о возникновении ошибки в функции

nErrorId - Код возникшей ошибки

***Замечание:** Температура присваивается байту массива nDataBytes[1], а значение регулятора присваивается байту массива nDataBytes[2]. Температурные датчики имеют шкалу от 0°C до 40°C. Эта шкала линейна и растянута на 1 байт (0°C = 0, 40°C = 255), чтобы получить значение температуры, необходимо создать функцию, которая будет пересчитывать значение байта обратно.*

***Замечание:** Температурные датчики передают информацию 1 раз в 10/100секунд (в зависимости от DIP переключателя), поэтому для них Watchdog нужно установить более 2минут (twatchdog = t#2m)*

Пример EnOcean: Функциональный блок FB_EnOceanSTM250



Функциональный блок FB_EnOceanSTM250 – извлекают данные из структуры «stEnOceanReceivedData» полученной ф.б. «FB_EnOceanReceive» и записывает в bState и bLearn.

bEnable – Бит включения/выключения функции

tWatchdog – Время, через которое должны обновляться данные от FB_EnOceanReceive

nTransmitterId – ID передатчика, для которого назначена функция

stEnOceanReceivedData – Данные, получаемые от FB_EnOceanReceive

bError – Бит, свидетельствующий о возникновении ошибки в функции

nErrorId - Код возникшей ошибки

bState – Состояние контакта реле на модуле STM250 (“1” если замкнут)

bLearn – “0” если нажата кнопка “Learn” на модуле STM250

Пример EnOcean: Определение типа EnOcean модуля и его ID

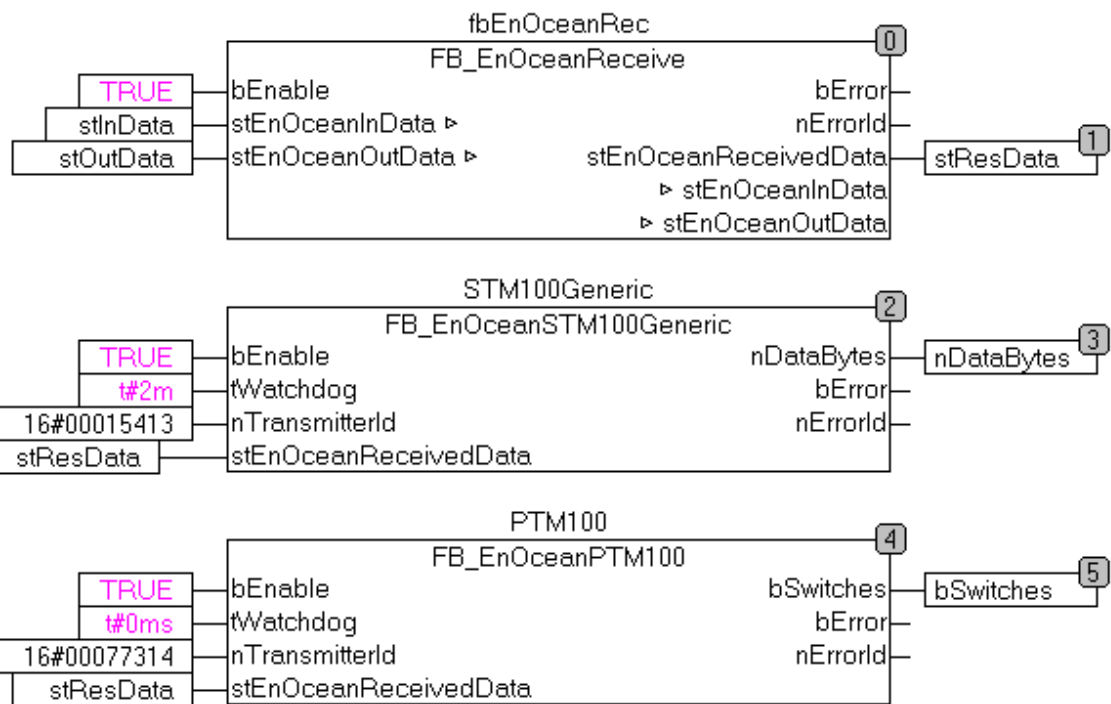
- **Определение типа:**

- 1) Прочитать в документации к устройству.
- 2) Прочитать на самом устройстве.

- **Определение ID:**

- 1) Прочитать в документации к устройству.
- 2) Прочитать на самом устройстве (не всегда).
- 3) Определить экспериментально (подробное описание на слайде 40)

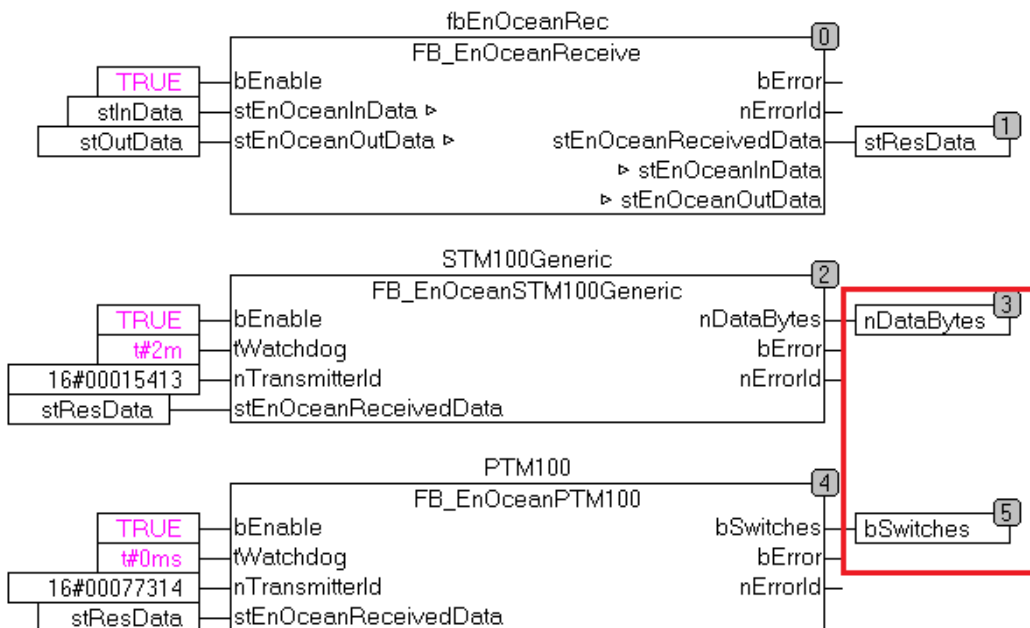
Пример EnOcean: связывание функциональных блоков



- Воспользуемся для нашего примера двумя функциональными блоками, которые были описаны выше.
- «**bEnable**» установим везде TRUE
- «**tWatchdog**» поставим исходя из замечаний со слайда 14.
- Для каждого EnOcean устройства которое мы используем нужно писать отдельный функциональный блок соответствующего типа и указывать в «**nTransmitterId**» ID блока в шестнадцатиричном виде, установленный одним из способов описанных выше.
- На вход «**stEnOceanReceivedData**» блоков указываем переменную полученную на выходе функционального блока «**FB_EnOceanReceive**»

Пример EnOcean: преобразование полученных данных

```
MAIN (PRG-CFC)
000 PROGRAM MAIN
000 VAR
000   fbEnOceanRec: FB_EnOceanReceive;
000   stInData AT %I*: ST_EnOceanInData;
000   stOutData AT %Q*: ST_EnOceanOutData;
000   stResData: ST_EnOceanReceivedData;
000   STM100Generic: FB_EnOceanSTM100Generic;
000   nDataBytes: ARRAY [0..3] OF BYTE;
000   bSwitches: ARRAY [0..7] OF BOOL;
001   PTM100: FB_EnOceanPTM100;
001 END_VAR
```



- Теперь необходимо создать переменные в которые будут записываться значения на выходе из функциональных блоков (В нашем случае массив из 4 значений типа **BYTE** и массив из 8 значений типа **BOOL**)
- Для этого в поле с описанием переменных (над основным полем программы), напишем следующие строки:
nDataBytes: ARRAY [0..3] OF BYTE;
bSwitches: ARRAY [0..7] OF BOOL;
- И теперь на выходе «nDataBytes» из блока «FB_EnOceanSTM100Generic» запишем переменную «nDataBytes».
- И на выходе «bSwitches» из блока «FB_EnOceanPTM100» запишем переменную «bSwitches».

Пример EnOcean: создание массива через Declare Variables

Declare Variable

Class: VAR, Name: nDataByts, Type: BOOL

Symbol list: Global_Variables

Initial Value: , Address:

Comment:

Buttons: OK, Cancel

Options: CONSTANT, RETAIN, PERSISTENT

Input assistant

Standard Types: ARRAY (selected)

User defined Types: BOOL, BYTE, DATE, DINT, DT, DWORD, INT, LREAL, REAL

Buttons: OK, Cancel

Array Boundaries

Dim.	Start	End
1	0	3
2		
3		

Type: BYTE

Buttons: OK, Cancel

Сделаем небольшое отступление и покажем как объявлять массивы через «**Declare Variables**»

Для этого как обычно вводим новое имя переменной в интересующем нас месте и затем в появившемся окне «**Declare Variable**» рядом с полем «**Type**» нажимаем «...»

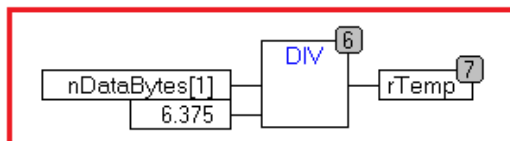
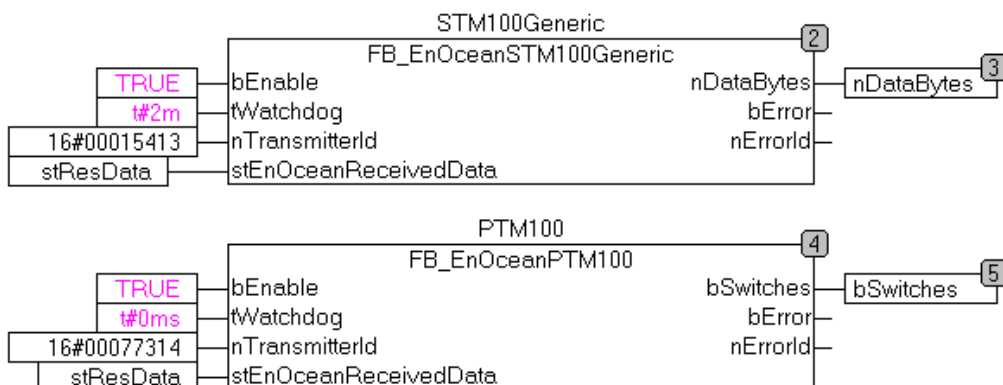
В появившемся окне «**Input Assistant**» выбираем «**ARRAY**»

В окне «**Array Boundaries**» указываем начало и конец массива по каждому измерению (в нашем случае оно одно) и выбираем нужный тип из списка нажимая «...», затем нажимаем «**OK**» и «**OK**».

Действия описанные по добавлению массива на этом слайде и предыдущем равнозначны (можете выбрать удобный для вас вариант).

Пример EnOcean: преобразование полученных данных

```
MAIN (PRG-CFC)
0001 PROGRAM MAIN
0002 VAR
0003   fbEnOceanRec: FB_EnOceanReceive;
0004   stInData: AT %I*: ST_EnOceanInData;
0005   stOutData: AT %Q*: ST_EnOceanOutData;
0006   stResData: ST_EnOceanReceivedData;
0007   STM100Generic: FB_EnOceanSTM100Generic;
0008   nDataBytes: ARRAY [0..3] OF BYTE;
0009   bSwitches: ARRAY [0..7] OF BOOL;
0010   PTM100: FB_EnOceanPTM100;
0011   rTemp: REAL;
0012 END_VAR
```



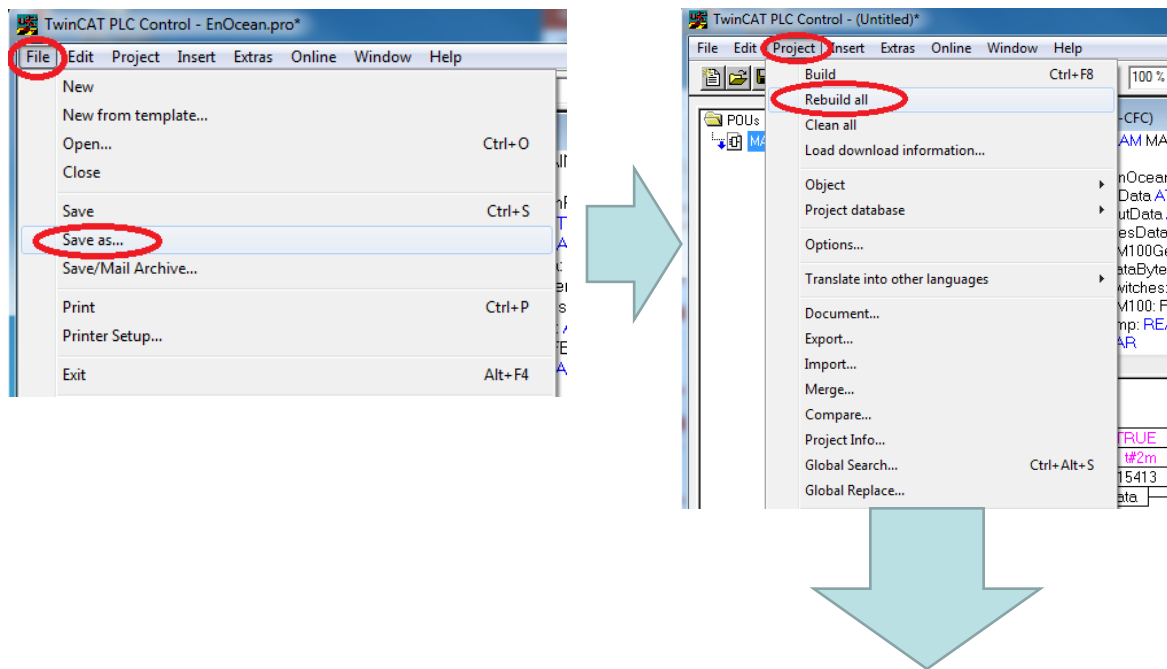
- Теперь напишем функцию позволяющую из нашего массива «nDataBytes» представить полученную температура в привычном виде
- Согласно замечанию описанному на слайде 15 для этого нужно значение «nDataBytes[1]» разделить на **6.375** и записать это в новую переменную «rTemp» типа «REAL».
- Для этого создаем блок как было описано выше, но с названием «DIV».

Пример EnOcean: связывание функциональных блоков.

- По факту мы полностью реализовали механизм получения данных с EnOcean модуля и можем использовать их в полной мере.
- Теперь разберемся как компилировать программу, конфигурировать контроллер, привязывать входа/выхода контроллера с входными и выходными переменными программы и запускать программу на контроллере.



Пример EnOcean: Компиляция программы



```
Implementation of POU 'MAIN'  
Implementation of task 'Standard'  
Warning 1990: No 'VAR_CONFIG' for 'MAIN.stInData'  
Warning 1990: No 'VAR_CONFIG' for 'MAIN.stOutData'  
POU indices:295 (14%)  
Size of used data: 1006 of 1048576 bytes (0.10%)  
Size of used retain data: 0 of 32768 bytes (0.00%)  
0 Error(s), 2 Warning(s).
```

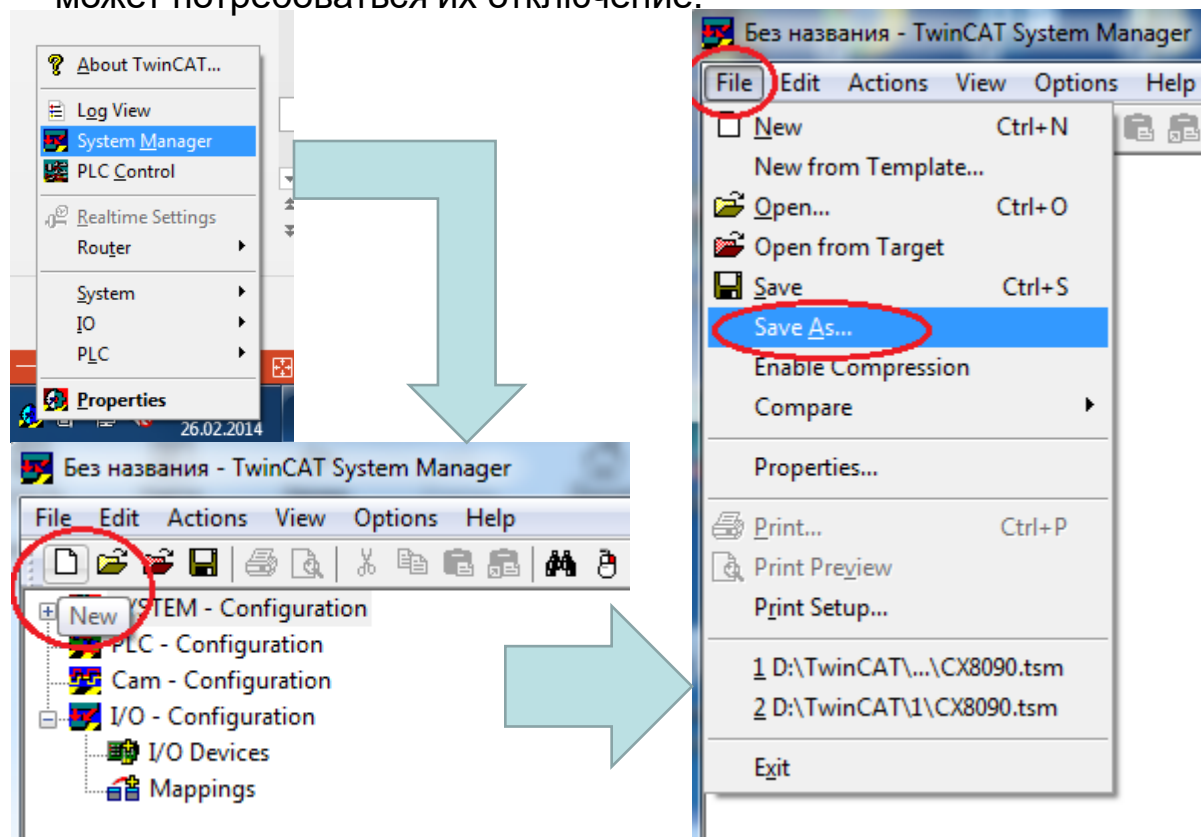
- Предварительно проект нужно сохранить: нажимаем на вкладку «**File**» и выбираем «**Save as...**», путь и название проекта нужно указывать **без символов кириллицы**.
- Для компиляции программы нажимаем на вкладку «**Project**» и выбираем пункт «**Rebuild all**»
- После компиляции программы, если все сделано верно, в поле системных сообщений не должно возникать никаких **Error**'ов и **Warning**'ов кроме оповещающих об отсутствии привязки входных/выходных переменных к аппаратной части контроллера.

Пример EnOcean: Добавление контроллера в AMS-роутер

Для того, чтобы подключиться к контроллеру необходимо выполнение следующих условий:

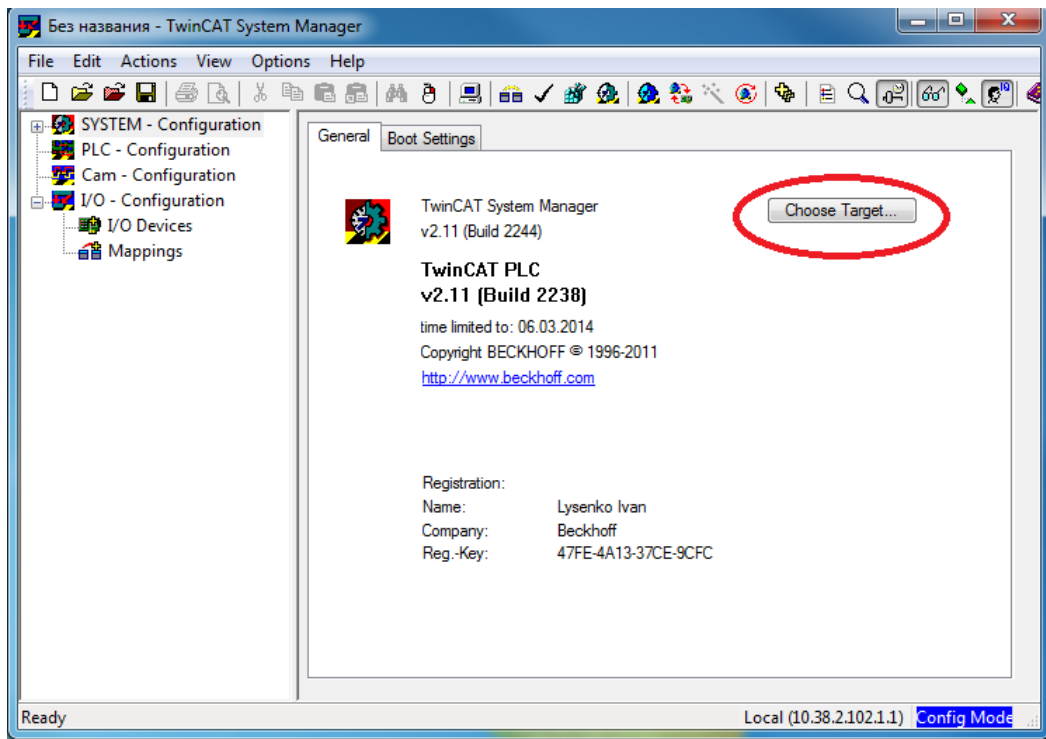
- 1) Работа на ПК из-под учётной записи с правами локального администратора
- 2) Нахождение в одной подсети IP-адресов с контроллером

В некоторых случаях Файерволы и Антивирусы могут блокировать прохождение TCP/IP пакетов, поэтому может потребоваться их отключение.

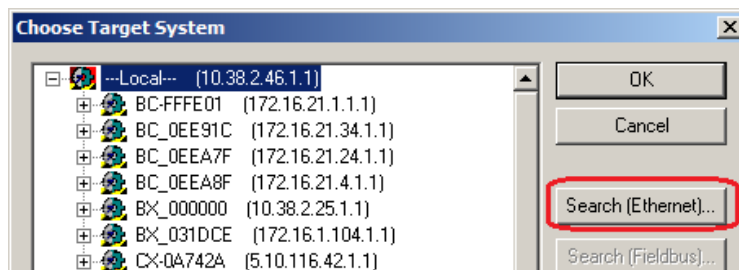


- Правой кнопкой мыши кликаем на значек TwinCAT в правом нижнем углу и выбираем “**System Manager**”
- Кликаем на значок «**New**» для создания нового проекта.
- И рекомендуем сразу сохранить проект: открываем вкладку «**File**» и выбираем «**Save as**», *для корректной работы необходимо, чтобы путь и название проекта не содержали символом кириллицы.*

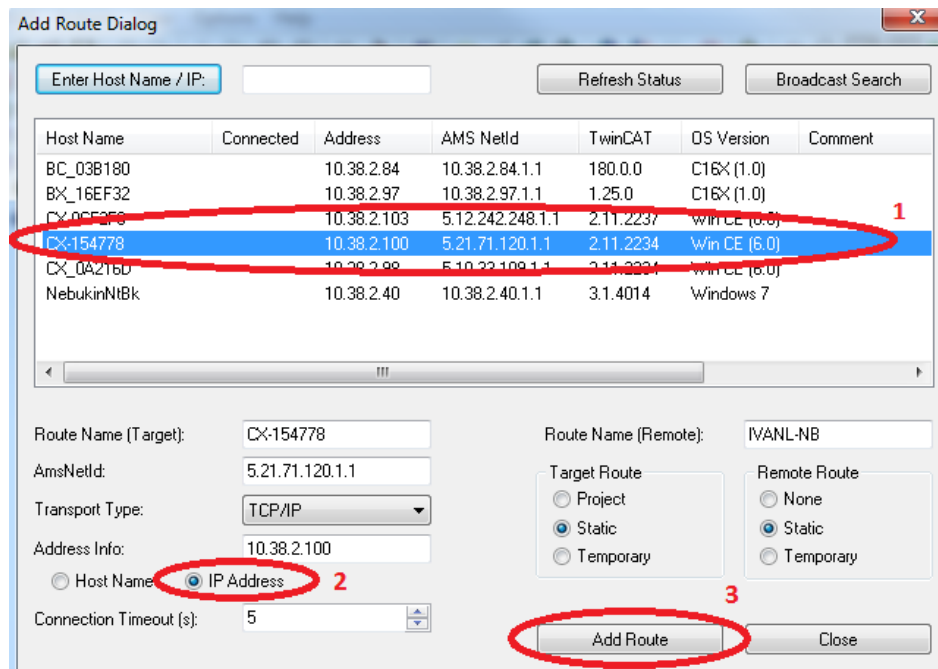
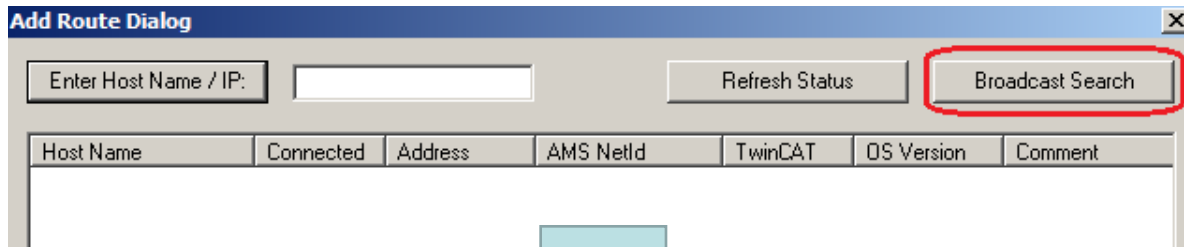
Пример EnOcean: Добавление контроллера в AMS-роутер



- В основном окне выбираем «**Choose Target...**»
- Выбираем «**Search (Ethernet)...**»

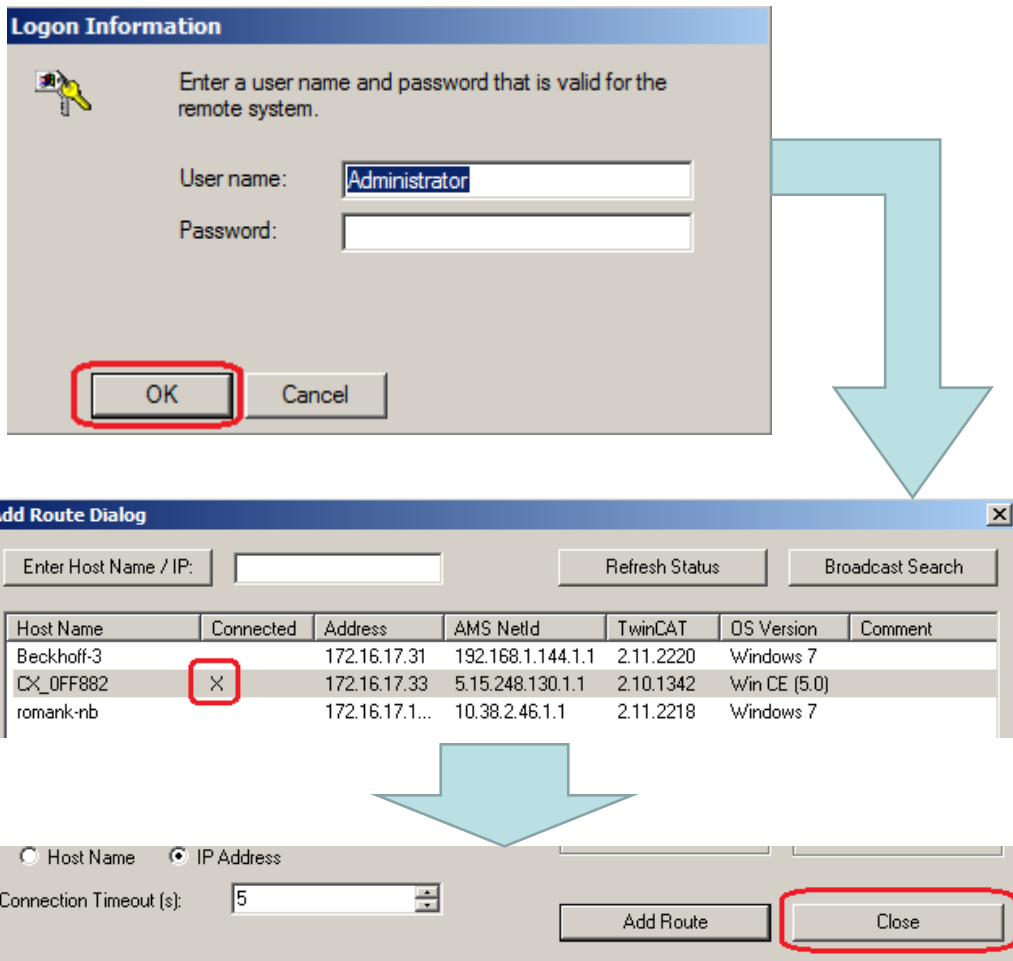


Пример EnOcean: Добавление контроллера в AMS-роутер



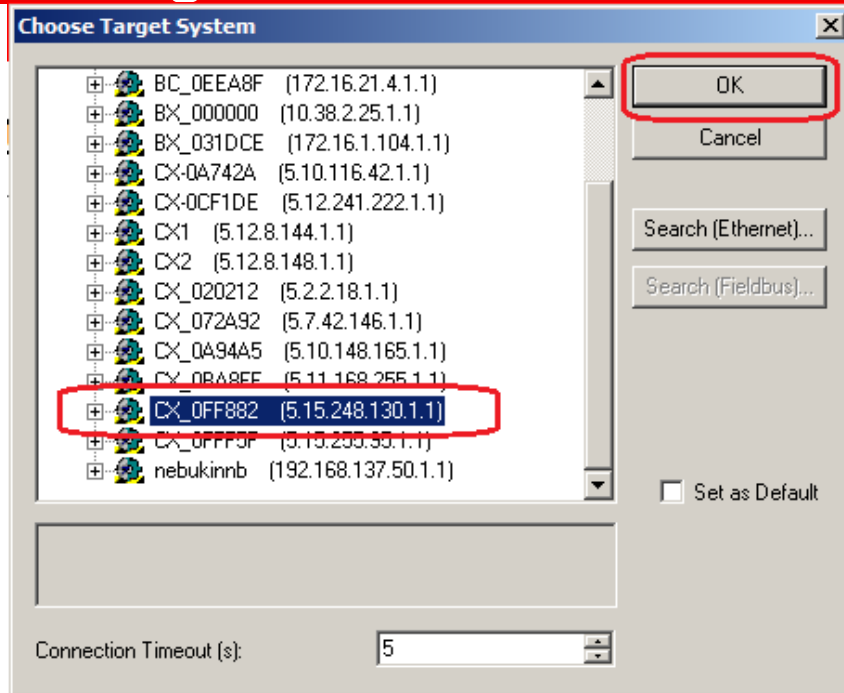
- Выполняем «**Broadcast Search**».
- Затем выбираем нужный контроллер
- Указываем привязку по IP адресу
- Нажимаем «**Add Route**»

Пример EnOcean: Добавление контроллера в AMS-роутер

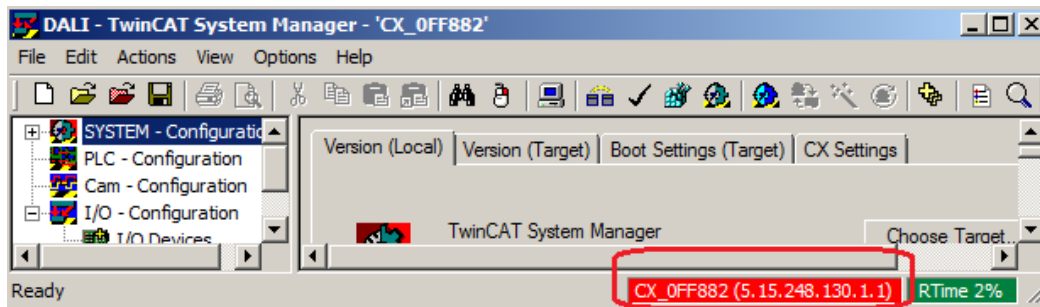


- Требуется авторизация под учётной записью пользователя операционной системы контроллера. В данном случае пароль не установлен, поэтому поле пароля остаётся пустым.
- Нажимаем «**OK**»
- Если произошло удачное добавление контроллера в AMS-роутер ПК разработчика, то в колонке “**Connected**” появляется значок “**X**”.
- Закрываем данное окно кнопкой «**Close**»

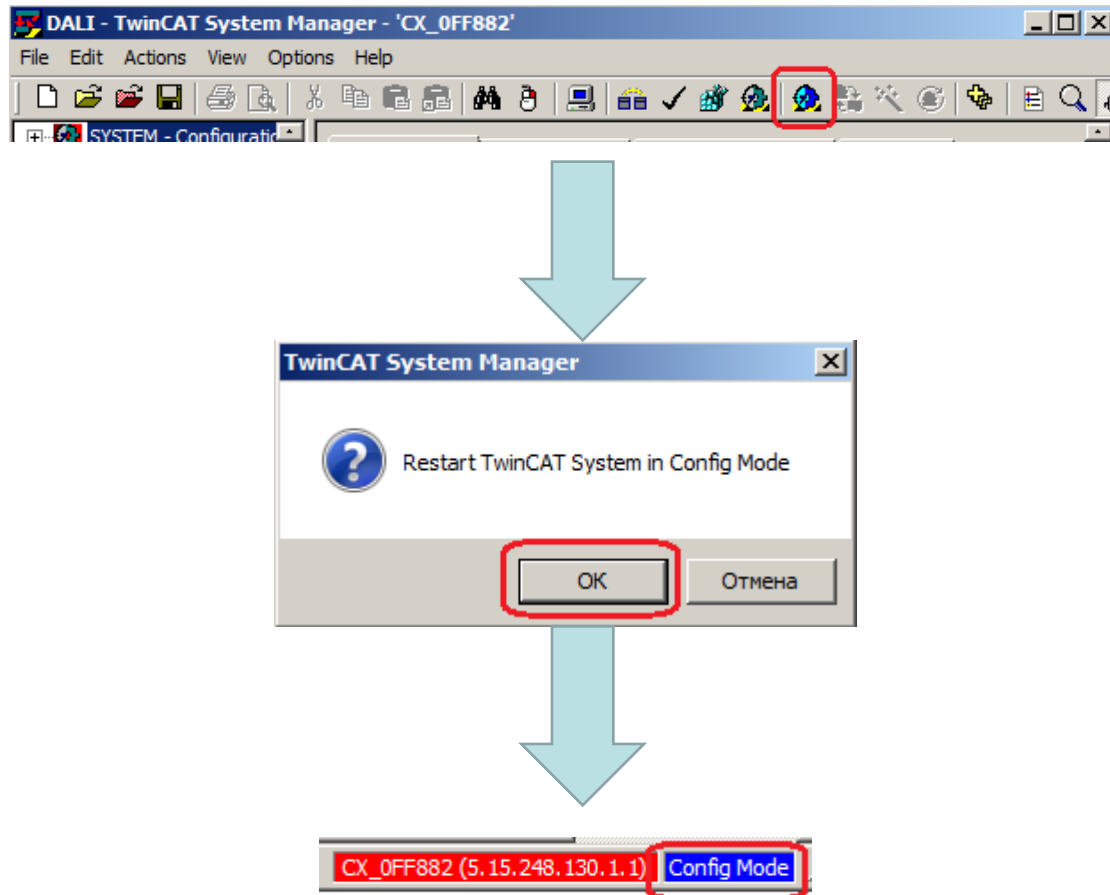
Пример EnOcean: Выбор контроллера в TwinCAT System Manager



- В окне «**Choose Target System**» выбираем контроллер и нажимаем «**OK**»
- После выбора контроллера, в нижнем правом углу System Manager должно отображаться его имя и режим.

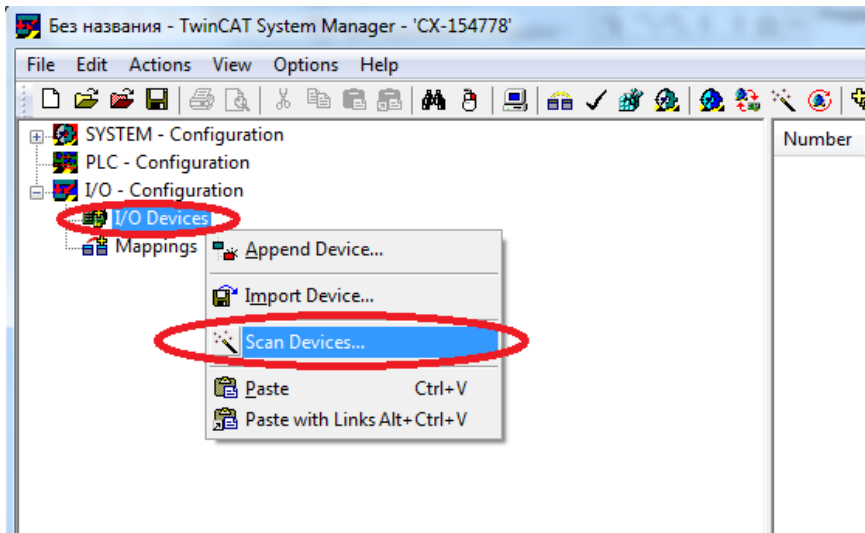


Пример EnOcean: Перевод контроллера в конфигурационный режим

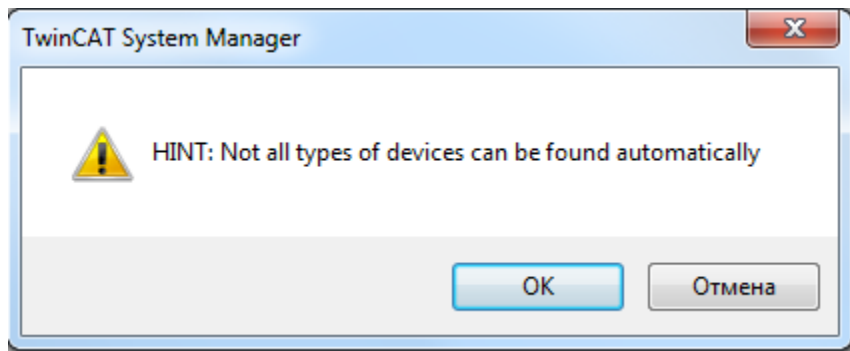


- Для дальнейшей конфигурации необходимо перевести контроллер в конфигурационный режим.
- Для этого жмем на указанный синий значек.
- Всплывает окно с запросом подтверждения, жмем «OK»
- Если все правильно, то справа внизу появится надпись на синем фоне «**Config Mode**»

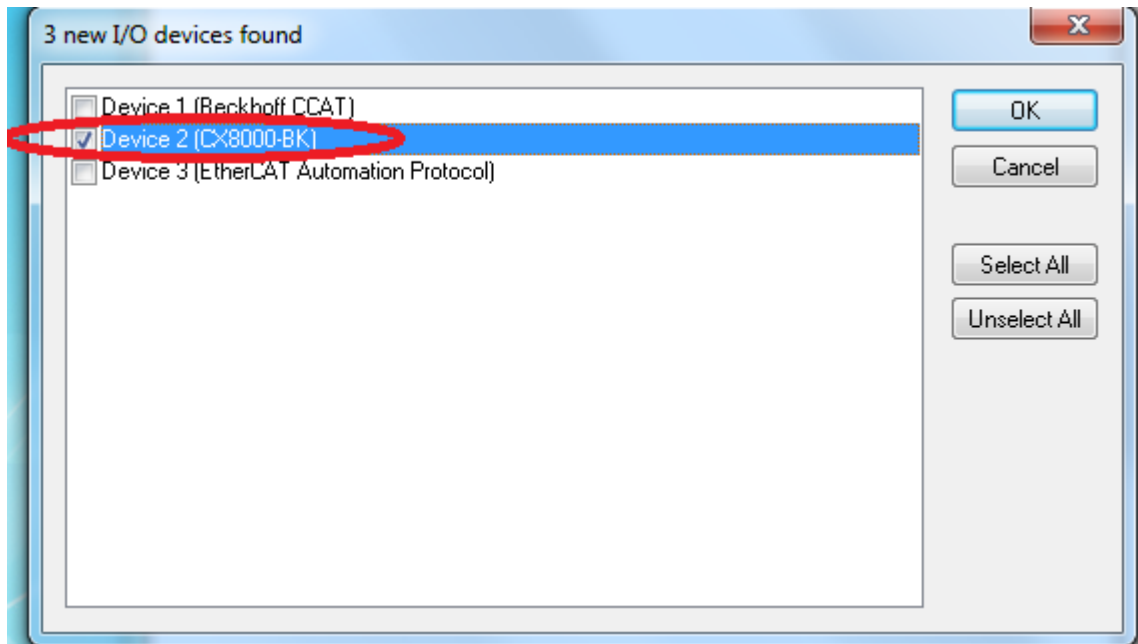
Пример EnOcean: Сканирование аппаратной части



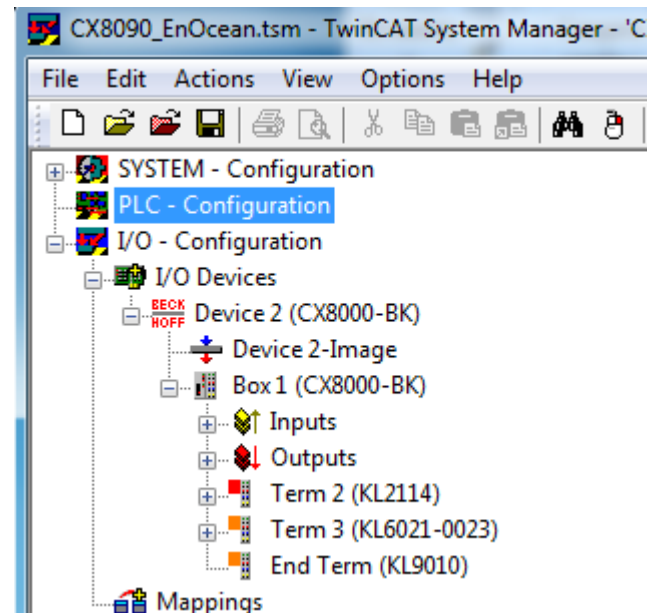
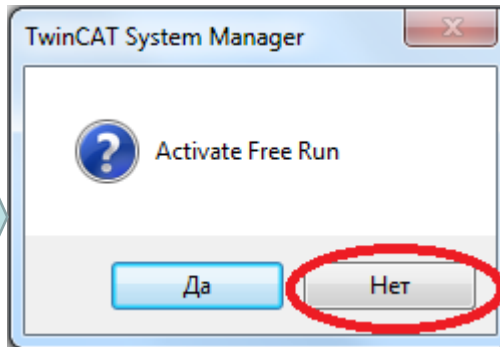
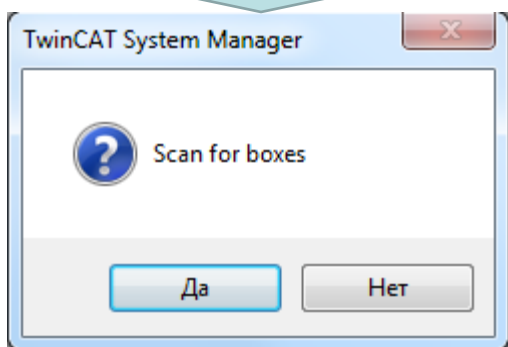
- Открываем дерево «**I/O – Configuration**» и кликаем правой кнопкой мыши по «**I/O Devices**» в открывшемся меню выбираем «**Scan Devices...**»
- На предупреждение отвечаем «**OK**»



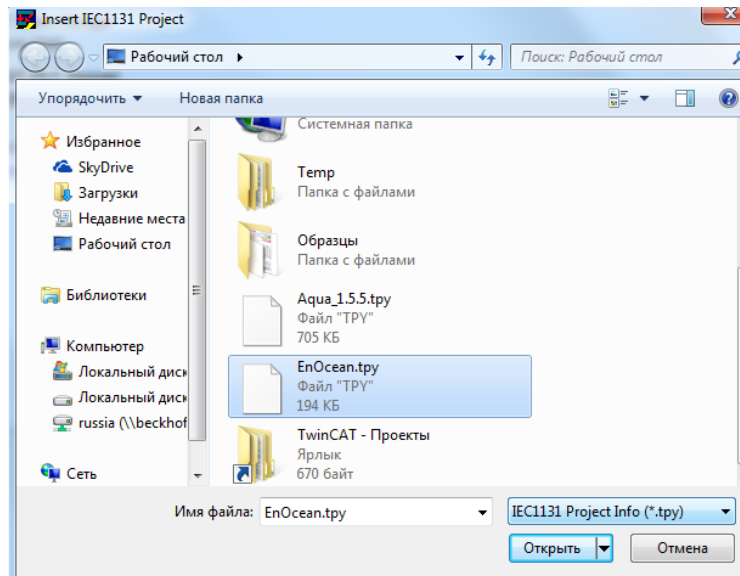
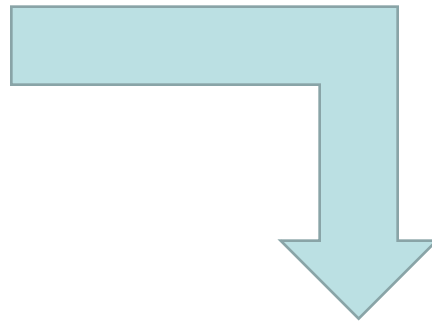
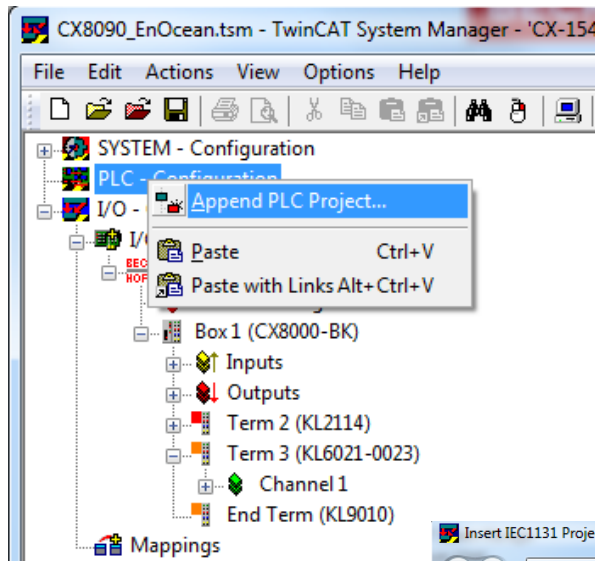
Пример EnOcean: Сканирование аппаратной части



- Выбираем нужный девайс.
- На вопрос «**Scan for boxes?**» отвечаем «**OK**»
- На вопрос «**Activate Free Run?**» отвечаем «**Нет**»



Пример EnOcean: Добавление PLC программы в TwinCAT System Manager



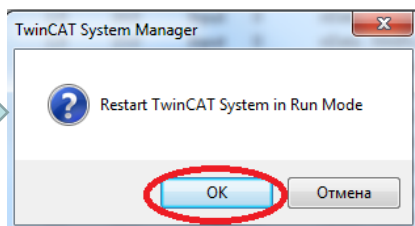
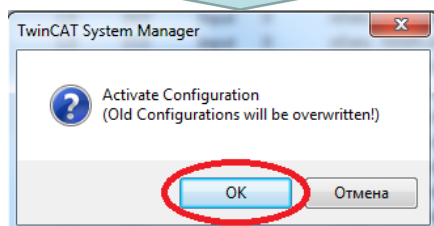
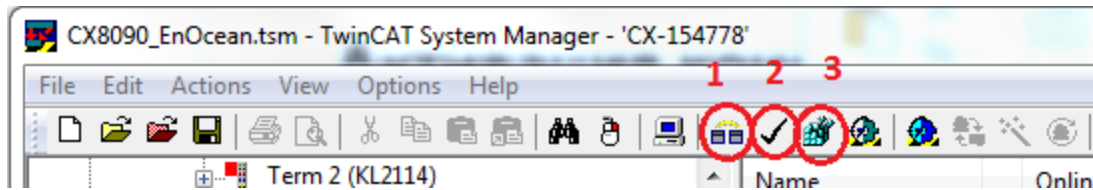
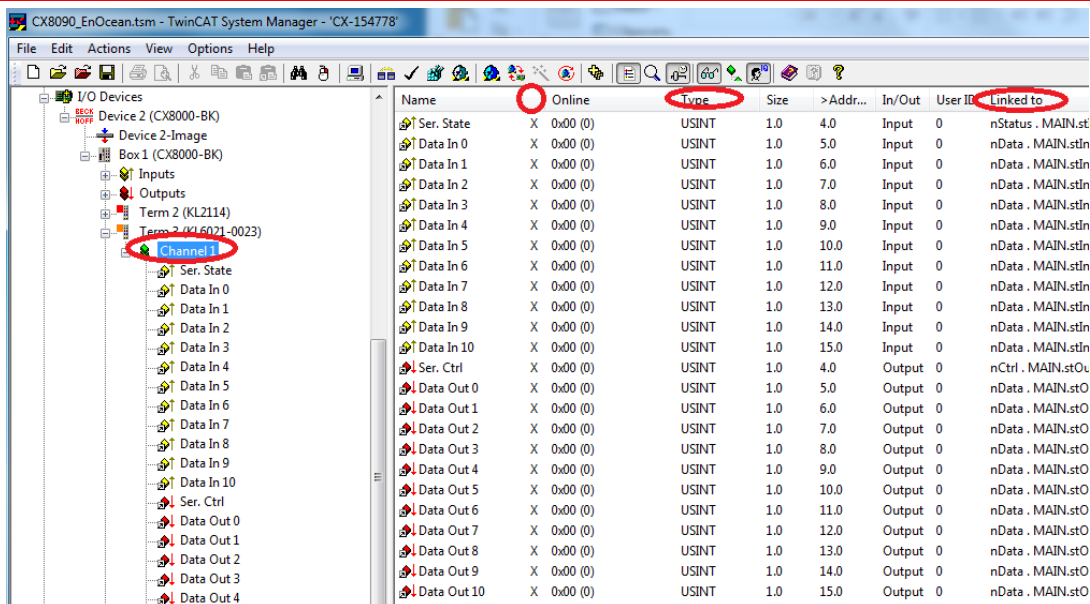
- Кликаем правой кнопкой мыши на дерево «**PLC - configuration**» и выбираем «**Append PLC Project...**»
- Выбираем ранее скомпилированный нами файл с названием проекта формата *.tpy и нажимаем «**Открыть**»
- Результатом добавления ПЛК-программы станет появление в узле PLC-Configuration подчинённого узла, содержащего имена внешних переменных ПЛК-программы

Пример EnOcean: Привязка переменных программы к аппаратной части (Multi Link)

The screenshot displays the TwinCAT System Manager interface for a Beckhoff PLC configuration. On the left, the 'I/O - Configuration' tree shows the hierarchy: 'I/O Devices' -> 'Device 2 (CX8000-BK)' -> 'Device 2 Image' -> 'Box 1 (CX8000-BK)' -> 'Inputs' -> 'Term 3 (KL6021-0023)' -> 'Channel 1'. Under 'Channel 1', there is a 'Ser. State' variable and 'Data In 0' through 'Data In 10'. A context menu is open over the 'Data In 11' row, with 'Change Multi Link...' selected. Below the main window, the 'Attach Variable 11.0 Byte(s) (Input)' dialog is shown, with the variable 'nData' selected in the tree, and 'IB 1.0, ARRAY [0..10] OF BYTE [11.0]' circled. The 'OK' button is also circled.

- Теперь необходимо привязать переменные, для этого заходим в «I/O – Configuration» выбираем модуль KL6021-0023 и открываем дерево «Channel1», где видим список входов/выходов которые надо привязать
- Справа появится список переменных, выделяем однотипные переменные (Data In 0 – Data In 10), кликаем правой кнопкой мыши и выбираем пункт «Change Multi Link...», выбираем нужный массив и жмем «OK».
- Аналогичное можно сделать с переменными Data Out 0- Data Out 10.

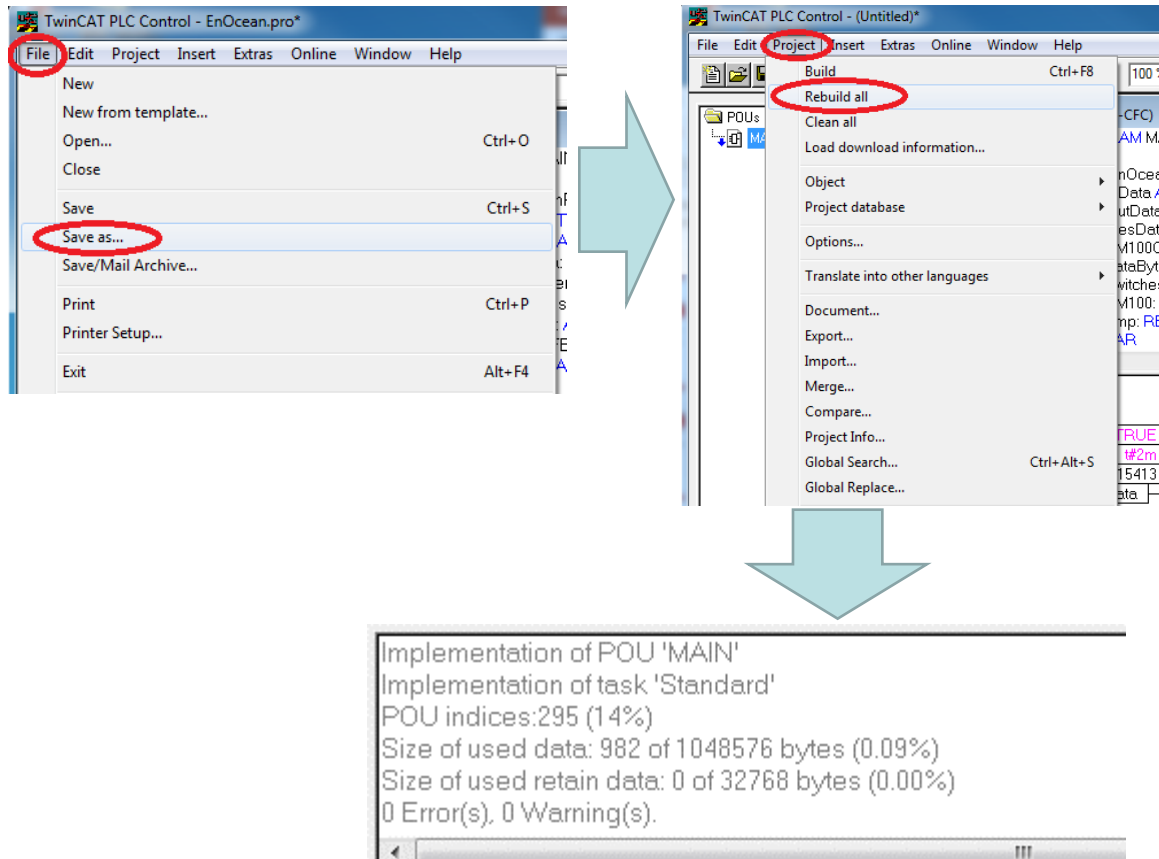
Пример EnOcean: Привязка переменных программы к аппаратной части



- После привязки всех переменных «Channel 1» будет выглядеть следующим образом.
- Привязанные переменные будут с символом «X», будет указан их тип и с какой переменной из PLC программы осуществлена привязка
- После привязке всех переменных необходимо нажать
 - 1) «Generate mappings»
 - 2) «Check Configuration»
 - 3) «Activate Configuration»

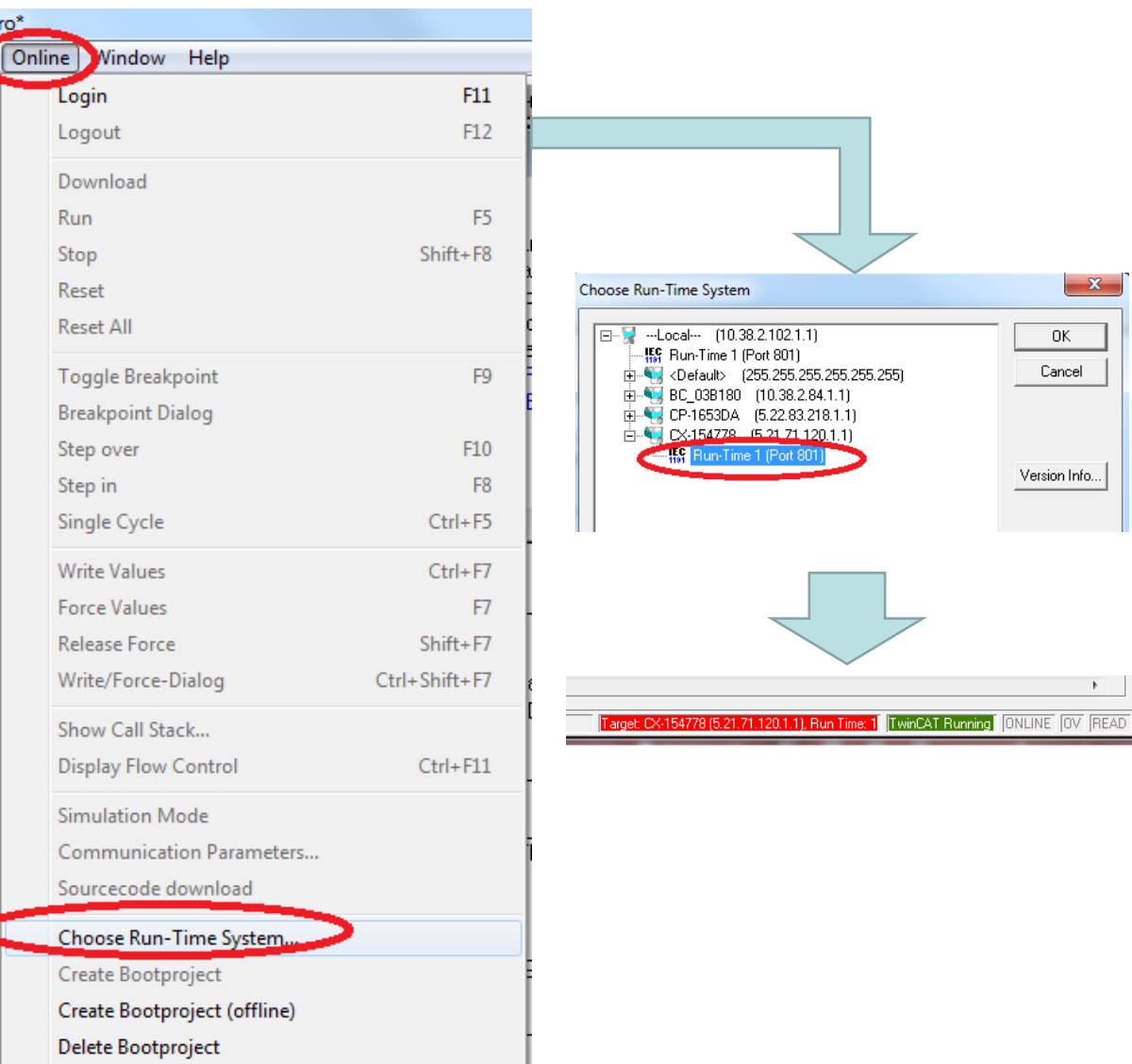
На всплывающих окнах нажимаем «OK»

Пример EnOcean: Повторная компиляция программы



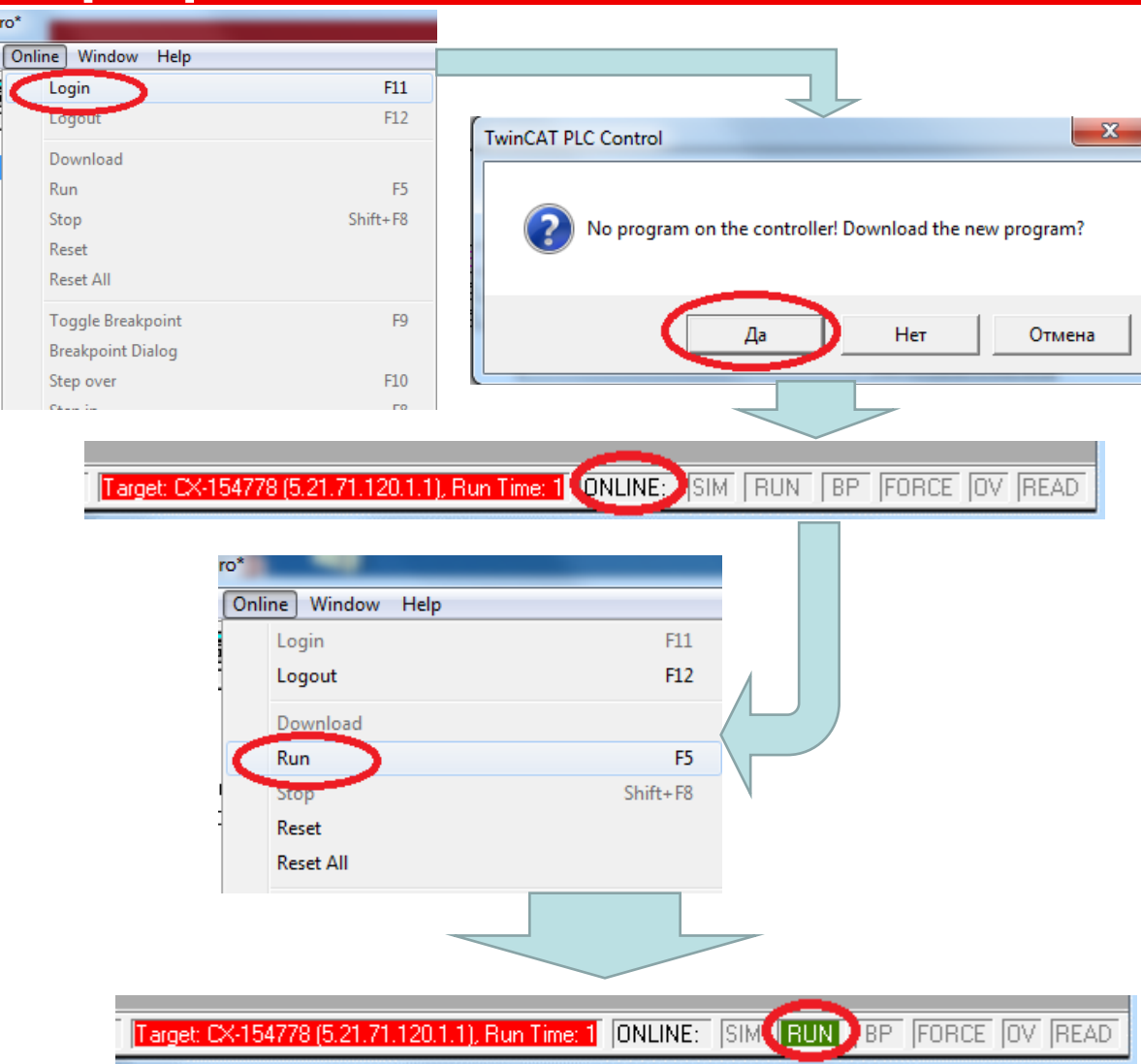
- После активации конфигурации из TwinCAT System Manager и перевода контроллера в режим “Run” нужно вернуться к проекту в TwinCAT PLC Control.
- Сначала выполняется “Rebuild All”, чтобы в ПЛК программе были учтены привязки переменных.
- На этот раз предупреждений об отсутствии линковки быть не должно.

Пример EnOcean: выбор контроллера для загрузки PLC-программы



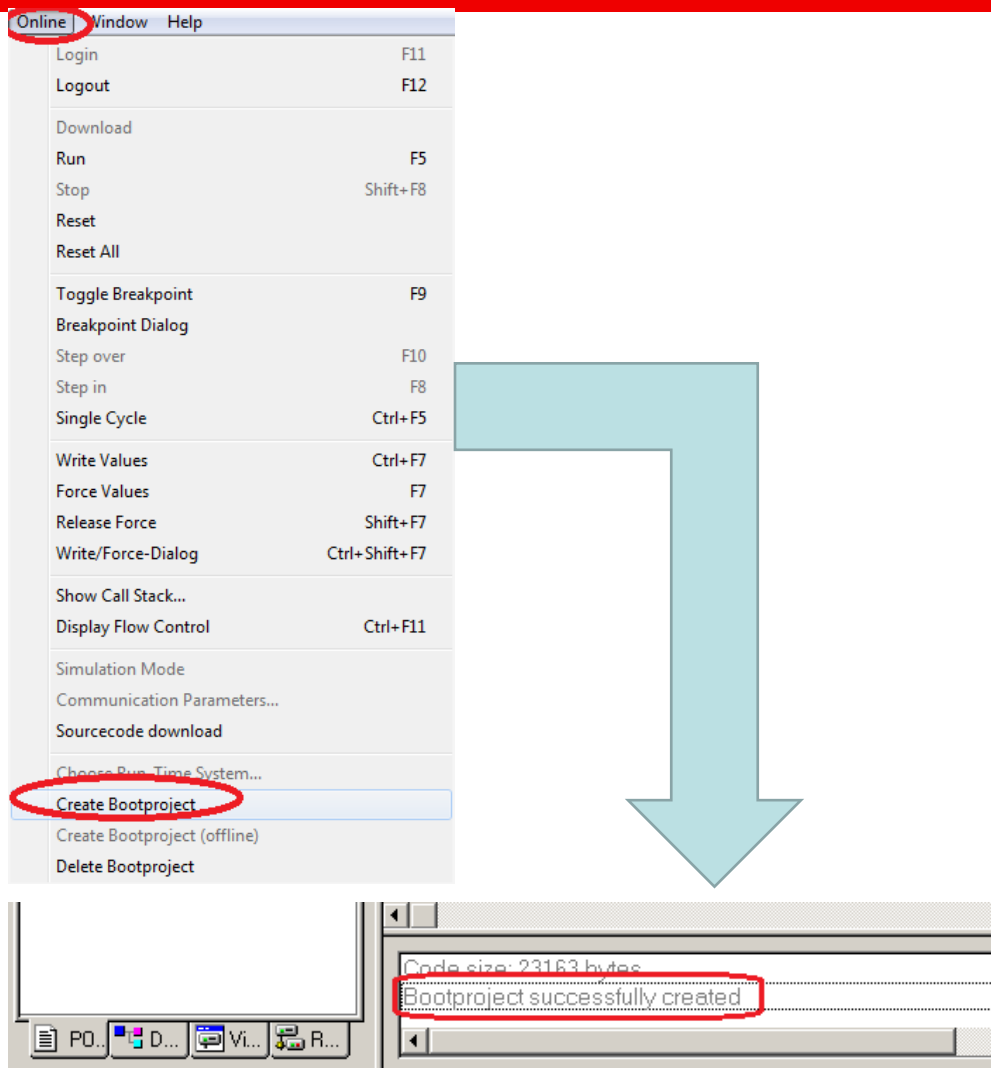
- Для загрузки программы в контроллер нужно кликнуть на вкладку «Online» и выбрать поле «Choose Run-Time System»
- Выбираем нужный контроллер, нужную Run-Time систему и нажать «OK»
- В нижнем правом углу TwinCAT PLC Control необходимо проверить:
 - 1) тот ли контроллер выбран?
 - 2) находится ли он в режиме Run?

Пример EnOcean: подключение к контроллеру и загрузка программы



- Для подключения к контроллеру нужно кликнуть на вкладку «Online» и выбрать поле «Login»
- Затем согласиться записать в контроллер новую программу или перезаписать имеющуюся в нем. Для этого в появившемся окне нажимаем «ДА»
- После подключения к контроллеру становятся видны текущие значения переменных ПЛК-программы, а надпись «ONLINE» в панели состояния становится контрастной
- Для запуска нужно кликнуть на вкладку «Online» и выбрать поле «Run», при этом надпись «Run» в панели состояние загорится зеленым.

Пример EnOcean: Создание загрузочного проекта



- После подключения к контроллеру Online и загрузки ПЛК-программы в него, эта программа храниться в **ОЗУ контроллера**. Когда программа отлажена и требуется, чтобы при включении контроллера она запускалась, нужно сохранить её в **ПЗУ контроллера**
- Для этого во вкладке «**Online**» выбираем пункт “**Create Boot Project**”
- Если загрузочный проект создан успешно, то в окне сообщений TwinCAT PLC Control появляется соответствующая строка.

Пример EnOcean: Экспериментальное определение ID для EnOcean устройства

```
MAIN (PRG-CFC)
0001 ⊖ fbEnOceanRec
0002   |---.nStep = 0
0003   |---.nType = 5
0004   ⊕ .nId
0005   |---.bEnable = TRUE
0006   |---.bError = FALSE
0007   |---.nErrorId = 0
0008   ⊖ stEnOceanReceivedData
0009   |---.bReceived = FALSE
0010   |---.nLength = 11
0011   |---.eEnOceanSensorType = eEnOceanSensorTypePTM
0012   ⊕ .nData
0013   |---.nStatus = 32
0014   |---.nTransmitterId = 1220104
0015   ⊕ .stEnOceanInData
0016   ⊕ .stEnOceanOutData
0017   ⊕ .stInData (%IB0)
0018   ⊕ .stOutData (%QB0)
0019   ⊕ .stResData
```

- Когда есть готовая программа, но не известны лишь ID модулей EnOcean их можно установить экспериментально.
- Для этого выполняем все действия описанные выше по запуску программы.
- Когда программа в режиме «Run» в поле переменных появятся значения всех переменных в текущий момент времени. Открываем блок соответствующий экземпляру «FB_EnOceanReceive», выбираем в нем «.stEnOceanReceivedData» и смотрим на переменную «.nTransmitterId», теперь нажимаем любую кнопку на интересующем нас устройстве EnOcean и в этом поле увидим его ID.

Пример EnOcean: заключение

В заключение можно сказать, что многие из описанных процедур являются весьма универсальными и применяются не только при работе с EnOcean устройствами.

Например: добавление различных блоков в PLC программу, связывание блоков, добавление переменных в программу, добавление контроллера в AMS-роутер, привязка аппаратной части, компилирование и методы отладки программы, подготовка контроллера, запуск программы на контроллере, создание загрузочного проекта и т.д.

Также многие приемы весьма удобны для проверки работоспособности контроллера, модулей контроллера, EnOcean модулей, связи контроллера, для отладки программ и т.д.

Образец программы (по которой строился весь пример) прилагается:



EnOcean.pro