

Данный проект демонстрирует пример использования библиотеки ModbusFB ([CODESYS Forge - MODBUS / Home / Home](#)) для настройки ПЛК NLScon-RSB в качестве ведомого устройства, а ПЛК Codesys Control Win V3 — в качестве ведущего в сети Modbus RTU. Для разработки проекта использована среда Codesys 3.5.16 patch 4 и установочный пакет компонентов для модулей RealLab! CoDeSys Linux package.

Обмен данными между ПЛК Codesys Control Win V3 и ПЛК NLScon-RSB осуществляется по сети Modbus RTU с помощью преобразователя интерфейсов NL-485-USB, подключенного к порту COM2 ПЛК NLScon-RSB и к порту USB компьютера, на котором запущена служба Codesys Control Win V3 (виртуальный ПЛК).

С помощью стандартного функционала Codesys в дерево устройств добавлены устройства MasterDevice (Codesys Control Win V3) и SlaveDevice (NLScon-RSB). (рис. 1).

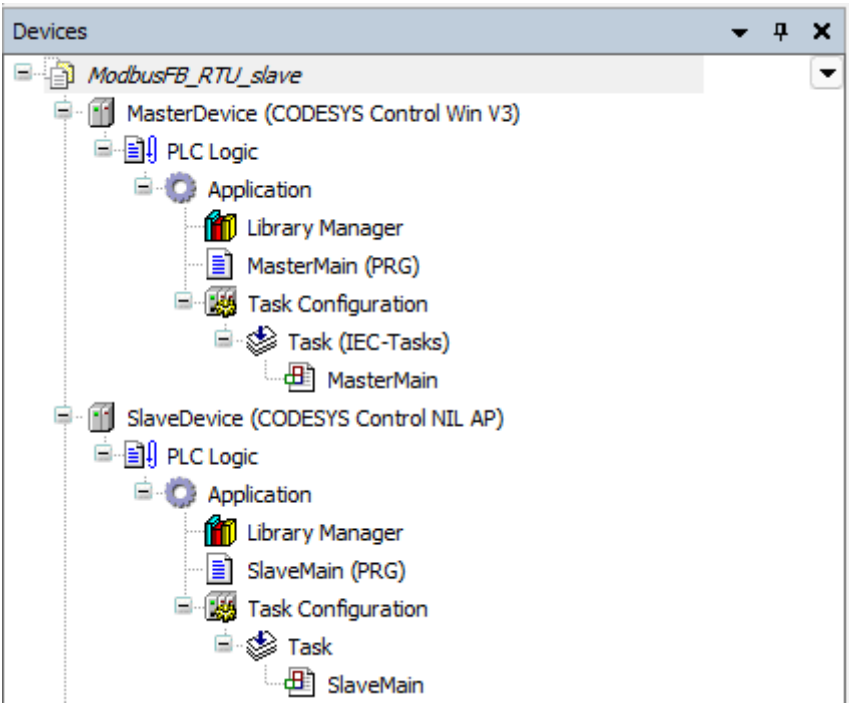


Рисунок 1 — Конфигурация дерева устройств для обмена данными в сети Modbus RTU

Для организации обмена с помощью функций библиотеки ModbusFB в Менеджере библиотек для ведущего и ведомого устройств добавлены библиотеки ModbusFB, SysCom и Standard (рис. 2).

Library Manager		
Add Library Delete Library Properties Details Placeholders Library Repository Icon Legend... Summary...		
Name	Namespace	Effective Version
3SLicense = 3SLicense, 3.5.16.0 (3S - Smart Software Solutions GmbH)	_3S_LICENSE	3.5.16.0
BreakpointLogging = Breakpoint Logging Functions, 3.5.5.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.5.0
CAA Device Diagnosis = CAA Device Diagnosis, 3.5.17.0 (CAA Technical Workgroup)	DED	3.5.17.0
IoStandard = IoStandard, 3.5.17.0 (System)	IoStandard	3.5.17.0
ModbusFB, 3.5.17.0 (3S - Smart Software Solutions GmbH)	ModbusFB	3.5.17.0
Standard, 3.5.15.0 (System)	Standard	3.5.15.0
SysCom, 3.5.15.0 (System)	SysCom	3.5.15.0

Рисунок 2 — Менеджер библиотек

Для настройки виртуального ПЛК Codesys Control Win V3 в качестве Modbus-мастера с помощью функционала библиотеки Modbus FB в программе MasterMain выполнено следующее:

1) создан экземпляр функционального блока (ФБ) мастера (клиента) ModbusFB.ClientSerial;

client : ModbusFB.ClientSerial;

2) созданы экземпляры ФБ для отправки запросов мастером:

- чтение регистров хранения (ФБ ModbusFB.ClientRequestReadHoldingRegisters);
- чтение входных регистров (ФБ ModbusFB.ClientRequestReadInputRegisters);
- запись регистров хранения (ФБ ModbusFB.ClientRequestWriteMultipleRegisters);

clientRequestReadHoldingRegisters : ModbusFB.ClientRequestReadHoldingRegisters;

clientRequestReadInputRegisters : ModbusFB.ClientRequestReadInputRegisters;

clientRequestWriteMultipleRegisters : ModbusFB.ClientRequestWriteMultipleRegisters;

3) созданы массивы данных для подготовки значений, записываемых в регистры хранения слева, и для хранения считанных из слева значений регистров хранения и входных регистров;

aUINT\_holding : ARRAY [0..10] OF UINT;

aUINT\_input : ARRAY [0..10] OF UINT;

aUINT\_multiple : ARRAY [0..10] OF UINT;

4) настройка мастера: настройка номера COM-порта, к которому подключен опрашиваемый слейв; скорости и формата передачи данных; протокол (RTU или ASCII); настройки вывода сообщений о ходе обмена данными в лог ПЛК. Настройки применятся по переднему фронту переменной xConnect

```
client(iPort:=SysCom.SYS_COM_PORTS.SYS_COMPORT3,  
      dwBaudrate := SysCom.SYS_BR_57600,  
      byDataBits:=byDataBits,  
      eRtuAscii:=eRtuAscii,  
      udiLogOptions:=(ModbusFB.LoggingOptions.ClientConnectDisconnect OR  
                      ModbusFB.LoggingOptions.ClientReceivedValidReplies));  
client(xConnect:=TRUE);
```

5) настройка параметров запросов: привязка запроса к мастеру; адрес слева; начальный адрес области данных; количество регистров; указатель на начало массива переменных мастера, куда будут записаны считанные из слева данные или где находятся подготовленные для записи в слейв значения

```
clientRequestReadHoldingRegisters(rClient:=client,  
                                  uiUnitId:=1,  
                                  uiStartItem:=5,  
                                  uiQuantity:=2,  
                                  pData:=ADR(aUINT_holding[0]));
```

6) после того, как мастер и запросы настроены, требуется циклически вызывать экземпляр client ФБ ModbusFB.ClientSerial и запускать запросы установкой их входных переменных xExecute в TRUE. Для проверки результатов работы запроса используются его

выходные переменные xDone и xError. Выход xDone = TRUE в случае успешно выполненного запроса, иначе xDone = FALSE, а выход xError = TRUE. После проверки результатов работы запроса следует сбросить переменную xExecute в FALSE.

Для настройки ПЛК NLScon-RSB в качестве Modbus-слейва с помощью функционала библиотеки Modbus FB в программе SlaveMain выполнено следующее:

1) создан экземпляр функционального блока (ФБ) мастера (клиента) ModbusFB.ClientSerial;

serverSerial: ModbusFB.ServerSerial;

2) созданы экземпляр ФБ ModbusFB.ExampleDataModel для формирования модели данных слейва — набора регистров, доступных для обмена данными по Modbus RTU.

exampleDataModel : ModbusFB.ExampleDataModel;

Модель данных ModbusFB.ExampleDataModel является примером формирования набора регистров для слейва и включает следующие секции:

- discrete inputs;
- coils;
- input registers;
- holding registers.

Каждая секция включает по 2 поля данных и имеет начальный адрес 5 по умолчанию.

3) настройка разрешенных для слейва Modbus-функций:

```
fcsSupported : ModbusFB.SupportedFcs := (  
    ReadHoldingRegisters:=TRUE,  
    ReadInputRegisters:=TRUE,  
    WriteMultipleRegisters:=TRUE  
);
```

4) настройка слейва: настройка разрешенных Modbus-функций, привязка к модели данных слейва, адрес слейва, настройка номера COM-порта слейва, к которому подключен мастер; скорость и формат передачи данных; протокол (RTU или ASCII); настройки вывода сообщений о ходе обмена данными в лог ПЛК. Настройки применяются по переднему фронту переменной xEnable

```
serverSerial(fcsSupported:=fcsSupported,  
    dataModel:=exampleDataModel.tableDefs,  
    xEnable:=FALSE,  
    uiUnitId:=1,  
    iPort:=SysCom.SYS_COM_PORTS.SYS_COMPORT2,  
    byDataBits:=byDataBits,  
    eRtuAscii:=eRtuAscii,  
    dwBaudrate := SysCom.SYS_BR_57600,  
    udiLogOptions:=(ModbusFB.LoggingOptions.ServerStartStop OR  
        ModbusFB.LoggingOptions.ServerReceivedValidRequests));
```

5) после того, как слейв настроен, требуется циклически вызывать экземпляр

serverSerial ФБ ModbusFB.ServerSerial.

Вместо встроенной в библиотеку ModbusFB модели данных ModbusFB.ExampleDataModel можно настроить пользовательскую модель данных типа ModbusFB.TableDefinitions. Этот тип представляет собой структуру, включающую описание секций для четырех типов данных: discrete inputs, coils, input registers, holding registers.

В данном примере пользовательская модель данных реализована для устройства SlaveDevice\_RegisterModel в программе SlaveMain (см. рис. 3).

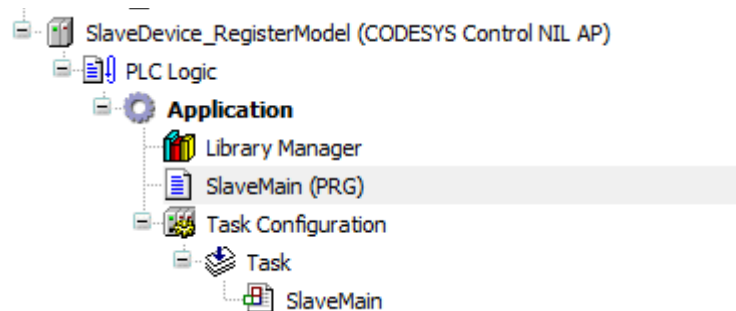


Рисунок 3 — Проект с пользовательской моделью данных в Modbus-слейв

В проекте создана пользовательская модель данных tableDefs типа ModbusFB.TableDefinitions (см. рис. 4). Эта модель в соответствии со своим типом состоит из четырех структур типа ModbusFB.TableDefinition:

- tableDiscreteInputs;
- tableCoils;
- tableInputRegisters;
- tableHoldingRegisters.

Каждая из перечисленных структур включает следующие поля: количество секций в структуре и указатель на первую секцию.

```
// ПОЛЬЗОВАТЕЛЬСКАЯ МОДЕЛЬ ДАННЫХ
tableDefs : ModbusFB.TableDefinitions := (
  tableDiscreteInputs := (
    uiNumSections := 1,
    pSections := ADR(aDiscreteInputsSections[0])
  ),
  tableCoils := (
    uiNumSections := 2,
    pSections := ADR(aCoilsSections[0])
  ),
  tableInputRegisters := (
    uiNumSections := 3,
    pSections := ADR(aInputRegistersSections[0])
  ),
  tableHoldingRegisters := (
    uiNumSections := 2,
    pSections := ADR(aHoldingRegistersSections[0])
  )
);
```

Рисунок 4 — Модель данных типа ModbusFB.TableDefinitions

Таким образом, для Modbus RTU слейва модель данных включает: 1 секцию «дискретных входов», 2 секции «катушек», 3 секции «входных регистров» и 2 секции «регистров хранения».

Секции задаются массивом из некоторого количества структур типа

ModbusFB.TableSection, например, инициализация секций для «входных регистров» показана на рис. 5.

```
aInputRegistersSections : ARRAY [0..2] OF ModbusFB.TableSection := [  
  // "input registers" 0..2  
  (  
    uiStart := 0,  
    uiNumDataItems := 3,  
    pStartAddr := ADR(aInputRegistersMemory1[0]),  
    uiDataItemSize := 16  
  ),  
  // "input registers" 5..8  
  (  
    uiStart := 5,  
    uiNumDataItems := 4,  
    pStartAddr := ADR(aInputRegistersMemory1[3]),  
    uiDataItemSize := 16  
  ),  
  // "input registers" 10..12  
  (  
    uiStart := 10,  
    uiNumDataItems := 3,  
    pStartAddr := ADR(aInputRegistersMemory2[0]),  
    uiDataItemSize := 16  
  )  
];
```

Рисунок 5 — Секция «входных регистров» типа ModbusFB.TableSection

В данном случае секции для «входных регистров» представляют собой 3 структуры типа ModbusFB.TableSection, которые описываются параметрами:

- начальный адрес uiStart;
- количество uiNumDataItems полей данных в секции (бит/байт/слов — в зависимости от назначения секции);
- указатель pStartAddr на область памяти в ПЛК, выделенную для этой секции;
- размер uiDataItemSize в битах поля данных. Для discrete inputs/coils значение uiDataItemSize может быть равно 1 или 8 бит, для input/holding registers значение uiDataItemSize равно 16 бит. Выбор размера для discrete inputs/coils зависит от того, как они привязаны к области памяти в ПЛК: если выделенная область памяти — это ARRAY [0..x] OF BOOL, то uiDataItemSize=8, если discrete inputs/coils адресуются побитово, то uiDataItemSize=1. Для input/holding registers область памяти должна быть массивом ARRAY [0..x] OF UINT.

В данном примере для секции «входных регистров» выделены области памяти:

```
aInputRegistersMemory1 : ARRAY [0..6] OF UINT;  
aInputRegistersMemory2 : ARRAY [0..2] OF UINT;
```

В нашем случае «входные регистры» для ПЛК настроены следующим образом:

- первая секция:
  - начальный адрес 0,
  - количество полей в секции 3 штуки (3 регистра),
  - их значения будут храниться в первых трех элементах массива aInputRegistersMemory1[0..2],
  - размерность каждого «входного регистра» — 16 бит.
- вторая секция:
  - начальный адрес 5,
  - количество полей в секции 4 штуки (4 регистра),

их значения будут храниться в последних четырех элементах массива `aInputRegistersMemory1[3..6]`,  
размерность каждого «входного регистра» — 16 бит.  
- третья секция:  
начальный адрес 10,  
количество полей в секции 3 штуки (3 регистра),  
их значения будут храниться в первых трех элементах массива `aInputRegistersMemory2[0..2]`,  
размерность каждого «входного регистра» — 16 бит.

Подобным образом в программе `SlaveMain` объявлены секции для `discrete inputs`, `coils` и `holding registers`. Для всех секций в программе должны быть выделены области памяти (см. рис. 6).

```
VAR
    // области памяти, которые ниже будут привязаны к секциям модели данных слейва
    aDiscreteInputsMemory : ARRAY [0..9] OF BOOL;
    aCoilsMemory1 : ARRAY [0..4] OF BOOL;
    aCoilsMemory2 : ARRAY [0..4] OF BOOL;
    aInputRegistersMemory1 : ARRAY [0..6] OF UINT;
    aInputRegistersMemory2 : ARRAY [0..2] OF UINT;
    aHoldingRegistersMemory : ARRAY [0..6] OF UINT;
END_VAR
```

Рисунок 6 — Области памяти для пользовательской модели данных ПЛК

Чтобы запустить Modbus RTU слейв на ПЛК, потребуется настроить `serverSerial` (экземпляр ФБ `ModbusFB.ServerSerial`) и разрешить его работу:

1) настройка слейва: настройка разрешенных Modbus-функций, привязка к модели данных слейва, адрес слейва, настройка номера COM-порта слейва, к которому подключен мастер; скорость и формат передачи данных; протокол (RTU или ASCII); настройки вывода сообщений о ходе обмена данными в лог ПЛК. Настройки применяются по переднему фронту переменной `xEnable`

```
serverSerial(fcsSupported:=fcsSupported,
    dataModel:=tableDefs,
    xEnable:=FALSE,
    uiUnitId:=1,
    iPort:=SysCom.SYS_COM_PORTS.SYS_COMPORT2,
    byDataBits:=byDataBits,
    eRtuAscii:=eRtuAscii,
    dwBaudrate := SysCom.SYS_BR_57600,
    udiLogOptions:=(ModbusFB.LoggingOptions.ServerStartStop OR
        ModbusFB.LoggingOptions.ServerReceivedValidRequests));
```

2) после того, как слейв настроен, требуется циклически вызывать экземпляр `serverSerial` ФБ `ModbusFB.ServerSerial`.

Для опроса слейва с пользовательской моделью данных, нужно в программе `MasterMain` заменить в настройках опроса начальный адрес и количество регистров (при необходимости), например, как на рис. 7:

```
clientRequestReadHoldingRegisters(rClient:=client, uiUnitId:=1, uiStartItem:=7, uiQuantity:=3, pData:=ADR(aUINT_holding[0]));  
clientRequestReadInputRegisters(rClient:=client, uiUnitId:=1, uiStartItem:=5, uiQuantity:=4, pData:=ADR(aUINT_input[0]));  
clientRequestWriteMultipleRegisters(rClient:=client, uiUnitId:=1, uiStartItem:=7, uiQuantity:=3, pData:=ADR(aUINT_multiple[0]));
```

Рисунок 7 — Задание начального адреса и количества регистров для опроса