

Данный проект демонстрирует пример использования библиотеки ModbusFB ([CODESYS Forge - MODBUS / Home / Home](#)) для настройки ПЛК NLScon-RSB в качестве ведущего устройства, а модуль NLS-16DI — в качестве ведомого 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 NLScon-RSB и модулем NLS-16DI осуществляется по сети Modbus RTU, модуль подключен к порту COM2 (1RS-485) ПЛК NLScon-RSB. Настройки модуля: адрес 9, скорость 57600 бит/с.

С помощью стандартного функционала Codesys в дерево устройств добавлено устройство MasterDevice (NLScon-RSB). (рис. 1).

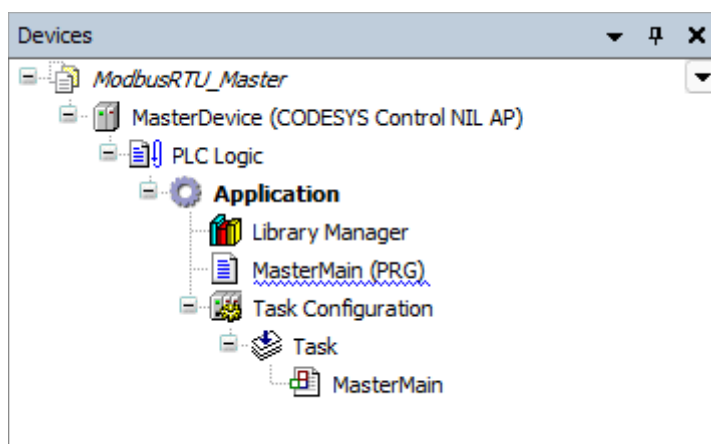


Рисунок 1 — Конфигурация дерева устройств для обмена данными в сети Modbus RTU

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 X 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 ПЛК NLScon-RSB 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);

```
clientRequestReadHoldingRegisters : ModbusFB.ClientRequestReadHoldingRegisters;  
clientRequestReadInputRegisters : ModbusFB.ClientRequestReadInputRegisters;
```

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

```
aUINT_holding : ARRAY [0..1] OF UINT;  
aUINT_input : ARRAY [0..1] 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 применяются по переднему фронту переменной xConnect

```
client(iPort:=SysCom.SYS_COM_PORTS.SYS_COMPORT2,  
      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 слейва; начальный адрес области данных; количество регистров; указатель на начало массива переменных мастера, куда будут записаны считанные из слейва данные или где находятся подготовленные для записи в слейв значения

```
clientRequestReadHoldingRegisters(rClient:=client,  
                                  uiUnitId:=9,  
                                  uiStartItem:=16#0200,  
                                  uiQuantity:=1,  
                                  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.

В данном примере используются два запроса: считать состояние дискретных входов (Modbus-функция 04, адрес 0h0000, 1 регистр) и считать адрес модуля (Modbus-функция 03, адрес 0h0200, 1 регистр).

В результате опроса модуля считанные данные записываются в соответствующие переменные в ПЛК (см. рис. 3): «входные регистры» - адрес 9, «регистры хранения» - состояние входов 4096 (0h1000 – 1 на входе Din12).

[-]	◆ aUINT_holding	ARRAY [0..1] OF UINT		
	◆ aUINT_holding[0]	UINT	9	
	◆ aUINT_holding[1]	UINT	0	
[-]	◆ aUINT_input	ARRAY [0..1] OF UINT		
	◆ aUINT_input[0]	UINT	4096	
	◆ aUINT_input[1]	UINT	0	

Рисунок 3 — Результат опроса модуля NLS-16DI