Данный проект демонстрирует пример использования библиотеки SysCOM для реализации обмена данными с внешним устройством через последовательный порт ПЛК NLScon-RSB (или NLScon-CE). В примере рассматривается возможность отправки запроса и приема ответа от устройства. В качестве подключенного к ПЛК устройства рассматривался модуль RealLab! NLS-16DI, настроенный для работы по протоколу DCON (57600, 8N1, адрес 09hex).

Для разработки проекта использована среда Codesys 3.5.16 patch 4 и установочные пакеты компонентов для модулей RealLab! CoDeSys Linux package.

Для работы в проект Codesys понадобится добавить библиотеку SysCOM. Эта библиотека реализует функции, выполняющие основные операции с COM портом: открытие, утсановка параметров соединнения, запись, чтение и закрытие (см. рис. 1).

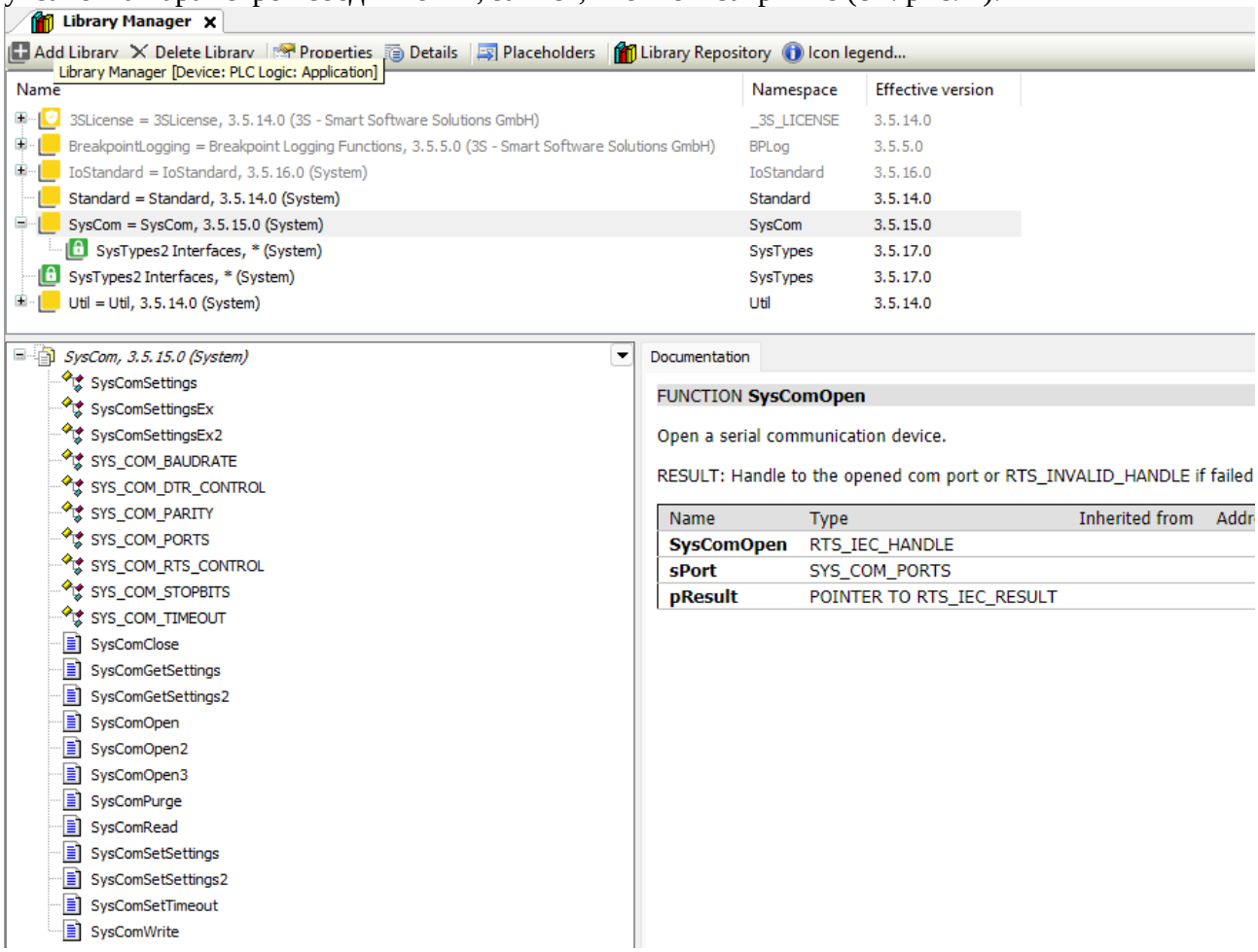

Рис.1 Библиотека SysCOM

В данном примере работа с COM-портом включает четыре этапа: открыте порта (и последующее задание настроек соединения), запись в порт, чтение ответа модуля NLS-16DI и закрытие порта.

Для открытия порта используется функция SysComOpen и затем, если получен обработчик порта, для задания типовых параметров соединения — функция SysComSetSettings (см. рис. 2).

```
// открыть COM port 1
hCom := SysComOpen(sPort:= SYS_COMPORT1 , pResult:= ADR(Result));
IF hCom <> RTS_INVALID_HANDLE THEN
    // настройка соединения
    csComSettings.byParity := SYS_NOPARITY;
    csComSettings.byStopBits := SYS_ONESTOPBIT;
    csComSettings.sPort := SYS_COMPORT1;
    csComSettings.ulBaudrate := SYS_BR_57600;
    csComSettings.ulBufferSize := 0;
    csComSettings.ulTimeout := 10;
    Result := SysComSetSettings(hCom:= hCom, pSettings:= ADR(csComSettings) , pSettingsEx:= 0);
END_IF
```

Рис.2 Открытие COM-порта

Таким образом, открытый порт настроен на передачу данных в COM порт №1 ПЛК со скоростью 57600 бит/с, формат 8N1.

Запись в порт производится с помощью функции SysComWrite (см. рис. 3). В функцию записи в качестве аргументов передается обработчик открытого порта, ссылка на начальный адрес подготовленных для записи данных, размер этих данных в байтах и количество милисекунд (ulTimeout), отведенных на запись. Функция возвращает количество отправленных в порт байтов. При этом, если за отведенное в ulTimeout время были переданы не все подготовленные для передачи байты, то соответственно возвращаемое количество байт будет меньше, чем ulSize.

```
// выполнить запись в порт
dwWritten := SysComWrite(hCom:= hCom,
                         pbyBuffer:= ADR(abyWriteData),
                         ulSize:= SIZEOF(abyWriteData),
                         ulTimeout:= 100,
                         pResult:= ADR(Result) );
```

Рис.3 Запись в COM-порт

После выполнения функции в переменную dwWritten запишется количество байт, отправленных в порт. В данном примере в порт отправляется последовательность символов: «^09Mcr» - команда чтения имени устройства по протоколу DCON.

Для чтения ответа модуля используется функция SysComRead. Агрументами для нее являются обработчик открытого COM-порта; ссылка на область данных для записи считанных байт; количество байт, которые предполаагается считать; количество милисекунд (ulTimeout), отведенных на чтение. Функция возвращает количество считанных за время ulTimeout байтов. При этом, если за отведенное в ulTimeout время были считаны не все байты, то соответственно возвращаемое количество байт будет меньше, чем ulSize.

```
// чтение данных из порта
dwRead := SysComRead(hCom:= hCom,
                     pbyBuffer:= ADR(strRead),
                     ulSize:= SIZEOF(strRead),
                     ulTimeout:= 1000,
                     pResult:= ADR(Result));
```

Рис.4 Чтение из COM-порта

Функция SysComClose закрывает COM-порт. Агрументом функции является обработчик открытого порта (см. рис. 5).

```
// закрыть порт
SysComClose(hCom:= hCom);
hCom := RTS_INVALID_HANDLE;
```

Рис.5 Закрытие COM-порта

В проекте используется функция _BUFFER_CLEAR() библиотеки OSCAT Basic 3.31 для очистки байтового массива для хранения считанных из COM порта данных.

Кроме того, в проекте использованы:

- генератор импульсов и R_TRIG для циклического запуска работы с портом:

```
bStart: BOOL := TRUE;
comStart : BLINK := (TIMELOW := T#500MS, TIMEHIGH := T#500MS);
bOpenCom : R_TRIG;

comStart(ENABLE:= bStart);
bOpenCom(CLK:= comStart.OUT);
```

- таймеры для реализации задержки в 50мс между этапами работы с портом:

```
tonWrDelay :TON := (PT:= T#50MS);
tonRdDelay :TON := (PT:= T#50MS);
tonClDelay :TON := (PT:= T#50MS);
```