



Описание интерфейса AMX-SCS (Realdom)  
между сетью SCS (MyHome-Bticino) и  
NetLinx Module (AMX)



## Содержание

История изменений.....	3
Введение .....	4
Общее описание .....	4
Использование.....	5
Интерфейс управления сетью SCS.....	7
Информация о состоянии сети SCS .....	9
Сообщения об ошибках .....	10
Конфигурация сети SCS.....	10
Замечания по программированию.....	11
Особенности реализации протокола.....	12

## История изменений

Дата	Комментарии
Февраль 2007	Разработка модуля и тестирование
Март 2007	Подготовка документации, создание серийного образца
Апрель 2007	Добавлена функция разбора строки, передаваемой от модуля в структуру пользователя

## **Введение**

В данном техническом руководстве содержится описание интерфейса между AMX Netlinx и сетью SCS (MyHome – VTicino) с использованием шлюза AMX-SCS и модуля VT\_TO\_AMX, разработанного компанией RealDom.

AMX модуль подсоединяется к сети SCS с использованием интерфейса AMX-SCS по 232 порту через стандартный интерфейсный кабель VTICINO артикула L4686 , имеющего следующие характеристики:

Baud rate = 38400 bps  
Data bits = 8  
Stop bit = 1  
Parity = none

Команда AMX, устанавливающая параметры порта должна выглядеть следующим образом:

```
send_string dvSCS_232, "SET BAUD 38400,N,8,1 485 DISABLE"
```

Кабель для подключения шлюза AMX-SCS по 232 порту (DB-9 Male) должен иметь следующий порядок разъемов:

### **РАСПАЙКА КАБЕЛЯ**

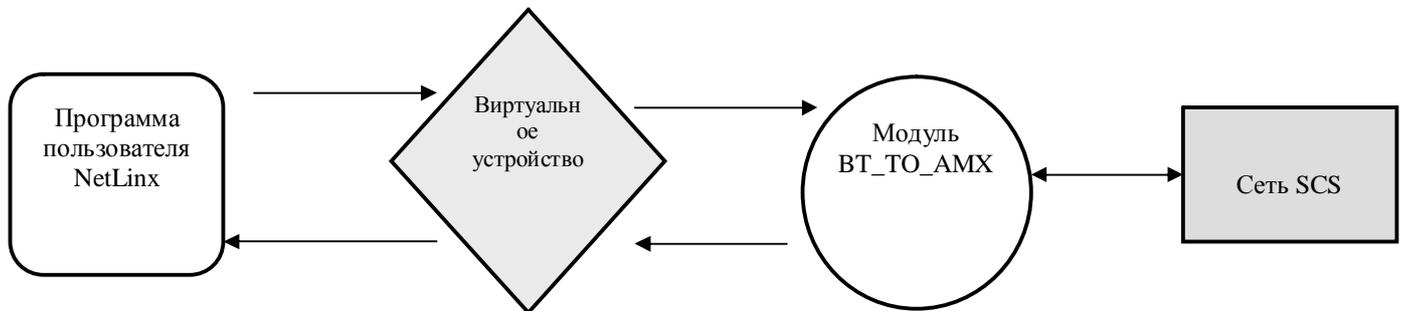
AMX NI	AMX-SCS
5 GND	5 GND
2 RXD	2 TXD
3 TXD	3 RXD

## **Общее описание**

Модуль VT\_TO\_AMX обеспечивает интерфейс между шлюзом AMX-SCS и программой пользователя NetLinx, выполняющейся на контроллерах AMX серии NI. Шлюз AMX-SCS обеспечивает согласование технических характеристик протоколов и уровней сигналов 232 порта процессора NI и интерфейсом 232 порта сети SCS, предварительный разбор протокола SCS перед передачей данных в модуль VT\_TO\_AMX, сжатие и подготовку данных для управления сетью SCS.

Модуль VT\_TO\_AMX является буфером между командами, поступающими из программы пользователя NetLinx для управления устройствами в сети SCS и строками данных, описывающих состояние сети SCS.

В графическом виде общую схему работы можно представить так:



Интерфейс BT\_TO\_AMX доступен пользователю только в виде модуля BT\_TO\_AMX.tko.

### Использование

Для использования BT\_TO\_AMX.tko модуля, программист должен выполнить следующие шаги:

1. Определить адрес порта к которому подключен шлюз AMX-SCS (например: COM-1).
2. Определить адрес виртуального устройства для связи с модулем BT\_TO\_AMX. Адреса таких устройств в NetLinx начинаются с 33001.
3. Определить адрес сенсорной панели для управления сетью SCS (**не является обязательным**).
4. Объявить модуль BT\_TO\_AMX.tko в программе NetLinx при помощи оператора DEFINE\_MODULE.
5. Подключить файл с описанием функций для работы с модулем BT\_TO\_AMX (#INCLUDE 'IncSCS.axi').

Пример, как это сделать приведен ниже.

```
DEFINE_DEVICE
dvSCS=5001:1:0           // Адрес порта RS232 к которому подключен шлюз AMX_SCS
SCS_Event=33001:1:0      // Виртуальное устройство для обмена с модулем BT_TO_AMX
dvTP = 10001:1:0        // Сенсорная панель управления

#include 'IncSCS.axi'     // Описание функций для работы с модулем BT_TO_AMX

DEFINE_VARIABLE
StSCS_Command SCS_Answer // Структура для выборки полученных команд пользователем
VOLATILE INTEGER dvTP_EVENT[] = // Коды клавиш на сенсорной панели для управления SCS
{
  101, // 1 - выполнить сценарий
  102, // 2 - управление активатором
  103, // 3 - регулировка яркости
```

```
104, // 4 - блокировка статуса
105, // 5 - GEN
106, // 6 - AMB
107 // 7 - GRP
}
```

```
DEFINE_MODULE 'BT_TO_AMX' SCS(dvSCS, SCS_Event) // Модуль - интерфейс между сетью
SCS и программой пользователя
```

## Интерфейс управления сетью SCS

Управление сеть SCS осуществляется при помощи функций описываемых ниже. Каждая функция преобразуется в команду для управления сетью SCS через модуль BT\_TO\_AMX (NetLinx command *send\_command*). Эти команды передаются модулю BT\_TO\_AMX для дальнейшей обработки и пересылке через шлюз AMX-SCS в сеть SCS.

Ниже приведен полный перечень функций, поддерживаемых модулем BT\_TO\_AMX.

Функция	Описание
BT_Scenario(SCS_Event,A,PL,KN)	<b>BT_Scenario</b> - Функция управления сценариями через MН200 SCS_Event - Адрес виртуального устройства NetLinx A – Зона (Area) PL - Номер устройства в зоне (Place of Light) KN – номер клавиши на устройстве <b>Пример:</b> BT_Scenario(SCS_Event,1,2,1)
BT_OODB(SCS_Event,A,PL,Status)	<b>BT_OODB</b> - Функция управления активаторами для включения и выключения OODB - ON/OFF/DIM/BRIGHT SCS_Event - Адрес виртуального устройства NetLinx A – Зона (Area) PL - Номер устройства в зоне (Place of Light) Status=1 OFF, Status=0 ON, Status=3 BRIGHT, Status=4 DIM <b>Пример:</b> BT_OODB(SCS_Event,1,2,1)
BT_SetLLight(SCS_Event,A,PL,LLevel)	<b>BT_SetLLight</b> - Функция для регулировки яркости освещения SCS_Event - Адрес виртуального устройства NetLinx A – Зона (Area) PL - Номер устройства в зоне (Place of Light) LLevel= от 1 до 9, соответственно от 10% до 90% <b>Пример:</b> BT_SetLLight(SCS_Event,1,2,5)
BT_SetBlockStatus(SCS_Event,A,PL,Status)	<b>BT_SetBlockStatus</b> - Функция для блокировки статуса устройства SCS_Event - Адрес виртуального устройства NetLinx A – Зона (Area) PL - Номер устройства в зоне (Place of Light) Status=1 Blocked, Status=0 UnBlocked <b>Пример:</b> BT_SetBlockStatus(SCS_Event,1,2,1)
BT_GEN(SCS_Event,Status)	<b>BT_GEN</b> - Функция для команды GEN SCS_Event - Адрес виртуального устройства NetLinx Status=1 OFF, Status=0 ON <b>Пример:</b> BT_GEN(SCS_Event,1)
BT_AMB(SCS_Event,PL, Status)	<b>BT_AMB</b> - Функция для команды GEN SCS_Event - Адрес виртуального устройства NetLinx PL - Номер устройства в зоне (Place of Light) Status=1 OFF, Status=0 ON <b>Пример:</b> BT_AMB(SCS_Event,2,1)

BT_GRP(SCS_Event,PL, Status)	<b>BT_GRP</b> - Функция для команды GRP SCS_Event - Адрес виртуального устройства NetLinx PL - Номер устройства в зоне (Place of Light) Status=1 OFF, Status=0 ON <b>Пример:</b> BT_GRP(SCS_Event,2,1)
------------------------------	---

**Таблица 1 – перечень используемых функций**

## Информация о состоянии сети SCS

Модуль VT\_TO\_AMX обеспечивает пересылку данных, генерируемых активаторами входящими в сеть SCS, в программу NetLinx.

Данные, пересылаемые в программу NetLinx поступают в виде строк (CHAR STR[7]) на виртуальное устройство SCS\_EVENT и могут быть обработаны через событие NetLinx:

```
DATA_EVENT[SCS_EVENT] {
```

```
    STRING: {
```

```
    // Переменная DATA.TEXT содержит массив из 7 (семи) символов
```

```
    }
```

```
}
```

Структура данных, принимаемых через переменную DATA.TEXT зависит от события, произошедшего в сети SCS, но общий формат остается неизменным и выглядит следующим образом:

Длина строки – 7 байт.

1	2	3	4	5	6	7
<b>\$F0</b>	xx	xx	xx	xx	xx	<b>\$F7</b>

Байт № 1 - **\$F0** – начало строки, полученной от сети SCS.

Байт № 2 - Тип устройства:

0 = нет команды,

1 = реле,

2 = диммер (только при включении устройства),

3 = датчик движения,

4 = выключатель как блок сценариев

Байт № 3 - Номер конфигуратора "A"

Байт № 4 - Номер конфигуратора "PL"

Байт № 5 - Байт статуса для диммера, датчика движения и сценария.

Байт № 6 - Байт команды для всех устройств.

Байт № 7 - **\$F7** – конец строки, полученной от сети SCS.

Для корректного разбора данных из полученной строки можно использовать специальную функцию 'GetSCS', описание которой находится в файле IncSCS.axi. Эта функция анализирует принятую строку (STR\_FROM\_SCS) и заполняет структуру (SCS\_Answer), объявленную пользователем в своей программе:

Структура SCS\_Answer имеет тип StSCS\_Command? Который описан в файле IncSCS.axi так:

```
STRUCTURE StSCS_Command
```

```
{
```

```
    integer SCS_TypeCommand      // Тип устройства:0=нет команды, 1=реле, 2=диммер,
```

```
    3=датчик движения,4=выключатель как блок сценариев
```

```
    integer SCS_Area            // Номер конфигуратора "A"
```

```

integer SCS_Place      // Номер конфигуратора "PL"
integer SCS_Status_1  // Статус
integer SCS_Status_2  // Команда
}

```

После вызова функции CALL 'GetSCS'(STR\_FROM\_SCS, SCS\_Answer) переменные структуры SCS\_Answer будут иметь следующие значения:

SCS\_Answer.SCS\_TypeCommand - Тип устройства:

- 0=нет команды,
- 1=реле,
- 2=диммер,
- 3=датчик движения,
- 4=выключатель как блок сценариев

SCS\_Answer.SCS\_Area - Номер конфигуратора "A"  
 SCS\_Answer.SCS\_Place - Номер конфигуратора "PL"  
 SCS\_Answer.SCS\_Status\_1 - Статус  
 SCS\_Answer.SCS\_Status\_2 - Команда

## **Сообщения об ошибках**

Сообщения об ошибках выводятся командой SEND\_STRING на устройство 0 (диагностика).

Модуль BT\_TO\_AMX анализирует следующие виды ошибок:

Сообщение об ошибке	Описание
SEND_STRING 0, "'SCS_ERROR(1) : BAD STRING FROM SCS EVENT' "	Строка, поступившая из модуля BT_TO_AMX не является корректной (нет символов начала и конца строки)
SEND_STRING 0, "'SCS_ERROR(2) : BAD STRING TO SCS EVENT' "	Строка, поступившая в модуль BT_TO_AMX не является корректной (нет символов начала и конца строки)
SEND_STRING 0, "'SCS_ERROR(3) : BAD COMMAND TYPE TO SCS EVENT' "	Строка, поступившая в модуль BT_TO_AMX содержит не верный (неподдерживаемый) код команды

Таблица 1 – Сообщения об ошибках

## **Конфигурация сети SCS**

Интерфейс AMX – SCS подключается к COM порту на контроллере AMX с одной стороны и к интерфейсу L4686 с другой.

В данный момент программное обеспечение интерфейса позволяет управлять сетью SCS в пределах одной логической зоны BTICINO. Если в системе две логических зоны, то необходимо объявить дополнительные устройства и модули, например:

## DEFINE\_DEVICE

```
dvSCS_1=5001:1:0 // Адрес порта RS232 к которому подключен шлюз
dvSCS_2=5001:2:0 // Адрес порта RS232 к которому подключен шлюз
AMX_SCS SCS_Event_1=33001:1:0 // Виртуальное устройство для обмена с модулем
BT_TO_AMX
AMX_SCS SCS_Event_2=33002:1:0 // Виртуальное устройство для обмена с модулем
BT_TO_AMX
```

## DEFINE\_VARIABLE

```
StSCS_Command SCS_Answer_1 // Структура для выборки полученных команд пользователем
```

## DEFINE\_VARIABLE

```
StSCS_Command SCS_Answer_2 // Структура для выборки полученных команд пользователем
```

```
DEFINE_MODULE 'BT_TO_AMX' SCS_1(dvSCS_1, SCS_Event_1) // Модуль - интерфейс между
сетью SCS и программой пользователя
```

```
DEFINE_MODULE 'BT_TO_AMX' SCS_2(dvSCS_2, SCS_Event_2) // Модуль - интерфейс между
сетью SCS и программой пользователя
```

## Замечания по программированию

Функции, используемые для управления сетью SCS (Таблица 1) преобразуют передаваемую им информацию в строку, пересылаемую далее командой SEND\_COMMAND виртуальному устройству, определенного пользователем для связи с модулем BT\_TO\_AMX.

Формат строки следующий:

Длина строки – 7 байт.

1	2	3	4	5	6
<b>\$F0</b>	xx	xx	xx	xx	<b>\$F7</b>

Байт № 1 - **\$F0** – начало строки, передаваемой в сеть SCS.

Байт № 2 – Тип команды:

- SCS\_BT\_Scenario - Функция управления сценариями через MH200
- SCS\_BT\_OODB - Функция управления активаторами для включения и выключен
- SCS\_BT\_SetLLight - Функция для регулировки яркости освещения
- SCS\_BT\_SetBlockStatus - Функция для блокировки статуса устройства
- SCS\_BT\_GEN - Функция для команды GEN
- SCS\_BT\_AMB - Функция для команды AMB
- SCS\_BT\_GRP - Функция для команды GRP

Байт № 3 - Номер конфигуратора “A” (если нет, то \$00)

Байт № 4 - Номер конфигуратора “PL” (если нет то \$00)

Байт № 5 – Статус или значение (зависит от типа команды):

Байт № 6 - **\$F7** – конец строки, передаваемой в сеть SCS.

## **Особенности реализации протокола**

В сетях SCS с большим количеством исполнительных устройств и активаторов в каждый момент времени возможно выполнение большого количества команд. С целью обеспечения их полной обработки в модуле VT\_TO\_AMX организован стек команд, заполняемый по мере их поступления по принципу FIFO (первый пришел, - первый ушел). Это позволяет передавать в программу пользователя все команды, генерируемые в сети SCS.