



Описание программы для контроллера АМХ.

Управление коммутаторами KRAMER по
протоколу Kramer Protocol 2000.



Содержание

История изменений.....	2
Введение	3
Общее описание	3
Использование.....	4
Пример использования.....	4
Запросы на получение данных.....	6
Команды управления.....	6
Подключение модуля к программе АМХ.....	7
Сообщения об ошибках	7

История изменений

Дата	Комментарии
Апрель 2006	Разработка модуля и тестирование

Введение

В данном техническом руководстве содержится описание программы для контроллера АМХ. Целью работы программы является управление коммутаторами Kramer, поддерживающими протокол Kramer Protocol 2000.

Компания Kramer предлагает очень широкую серию высококачественных и технологичных коммутаторов класса «HiEnd».

Для управления этими устройствами разработан специальный протокол Kramer Protocol 2000.

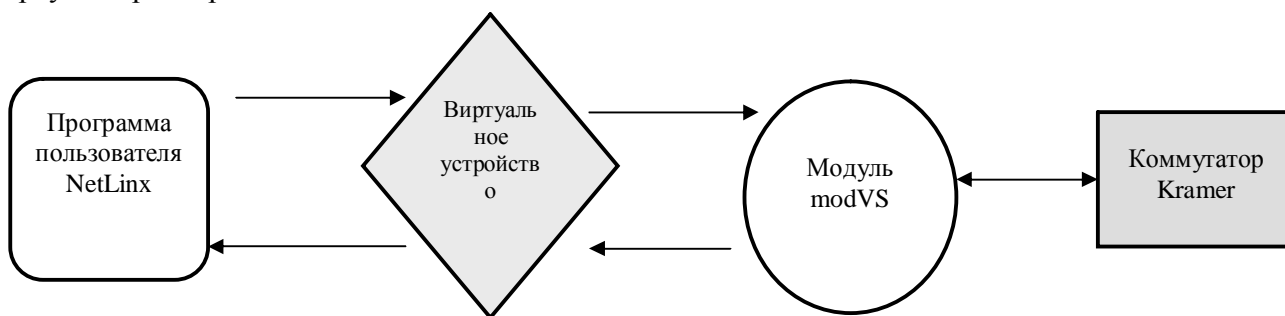
Мы не ставили себе задачу реализаций всех имеющихся возможностей, заложенных разработчиками в этот протокол (это возможно, при наличии соответствующих запросов от наших клиентов в самые короткие сроки), а реализовали самые востребованные в автоматизации функции (переключение входов на соответствующие выходы коммутата и информацию о текущем состоянии устройства).

Представленная программа для контроллера АМХ может работать как самостоятельно, так и в составе общего программного обеспечения создаваемого вами проекта «Умный дом». С этой целью, исполняемая часть оформлена в виде отдельного модуля легко адаптируемого в общее программное обеспечения.

Для подключения модуля необходимы начальные знания в программировании контроллеров АМХ.

Общее описание

Модуль modVS обеспечивает управление устройством Kramer VA-16x1 подключенного к 232/485 порту контроллера АМХ.



Модуль modVS является буфером между командами, поступающими из программы пользователя и устройством Kramer.

Модуль поддерживает возможность точечного (вход на выход) переключения (коммутации) каналов устройства, так и одновременного включения и выключения всех каналов одновременно по запросу из программы АМХ.

Настройка параметров порта 232/485 производится **автоматически** модулем modVSI.

Использование

Для использования modVS.tko модуля, программист должен выполнить следующие шаги:

1. Определить адрес виртуального устройства для обмена данными с модулем modVS.tko
Адреса таких устройств в NetLinX начинаются с 33001.
2. Определить порт процессора AMX (232/485) к которому подключено устройство Kramer.
3. Объявить модуль modVS.tko в программе NetLinX при помощи оператора DEFINE_MODULE.

Пример использования.

```
PROGRAM_NAME='KramerVS'
DEFINE_DEVICE
dvVS = 5001:1:0           // 232/485 порт на процессоре AMX, к которому подключено
устройство Kramer
vdvVS = 33001:1:0        // Виртуальное устройство для связи с модулем modVS
dvPanel = 10001:1:0      // Панель управления

DEFINE_CONSTANT
TUNER = 1                // FM tuner подключен к входу номер 1 коммутатора Kramer
KALEIDOS = 2            // Видеосервер подключен к входу номер 2 коммутатора Kramer
KITCHEN = 5              // Динамики на кухне подключены к выходу номер 5
BEDROOM = 3             // Телевизор и динамики в спальни подключены к выходу номер 3
Sound_TO_Kitchen = 1
Sound_AND_Video_TO_BedRoom = 2
ASK_Kitchen = 3
OFF_Kitchen = 4
OFF_ALL = 5

DEFINE_START
DEFINE_MODULE 'modVS' modKramerVS(dvVS, vdvVS) // Модуль работы с устройством
Kramer

DEFINE_EVENT

BUTTON_EVENT[dvPanel, Sound_TO_Kitchen] { // Включить радио на кухне
  PUSH: {
    SEND_COMMAND vdvVS, "INA=',ITOA(TUNER),';',ITOA(KITCHEN)"
  }
}
```

```

BUTTON_EVENT[dvPanel, Sound_AND_Video_TO_BedRoom] { // Включить звук и видео в
спальне
  PUSH: {
    SEND_COMMAND vdvVS, "IN=',ITOA(KALEIDOS),';',ITOA(BEDROOM)"
  }
}
BUTTON_EVENT[dvPanel, ASK_Kitchen] { // Узнать какое устройство «работает» на кухне
  PUSH: {
    SEND_COMMAND vdvVS, "IN?=',ITOA(KITCHEN)"
  }
}
BUTTON_EVENT[dvPanel, OFF_Kitchen] { // Выключить кухню
  PUSH: {
    SEND_COMMAND vdvVS, "IN=',ITOA(KITCHEN),';0"
  }
}
BUTTON_EVENT[dvPanel, OFF_ALL] { // Выключить все
  PUSH: {
    SEND_COMMAND vdvVS, "IN=0;0"
  }
}

DATA_EVENT[vdvVS] {
  STRING: {
    STACK_VAR
    CHAR tmpSTR[20]
    CHAR tmpCOMAND[20]

    tmpSTR = DATA.TEXT
    tmpCOMAND = REMOVE_STRING(tmpSTR, '=',1)
    SWITCH (tmpCOMAND) {
      CASE 'IN=: { // Сообщение состоянии выхода устройства
        tmpCOMAND = REMOVE_STRING(tmpSTR, ';',1)
        tmpCOMAND =
LEFT_STRING(tmpCOMAND,LENGTH_STRING(tmpCOMAND)-1) // Номер входа
        SEND_STRING 0,"INPUT=',tmpCOMAND,';OUTPUT=',tmpSTR" // Номер
входа и переключенного на него выхода
      }
    }
  }
}

```

Запросы на получение данных.

Получение данных о текущем состоянии выходов устройства Kramer возможно двумя способами:

1. Автоматически (если устройство изменяет свое состояние под воздействием внешней программы, нажатием кнопок на лицевой панели управления или по командам коммутации из программы на AMX - IN, INA, INV).
2. В любое время по запросу (по команде IN?=).

При изменении состояния устройства, модуль modVS обрабатывает эту информацию и посылает на устройство vdVS (см. пример) строку в формате:

IN=<Номер входа>;<Номер выхода>

Где, <Номер входа> - номер входа устройства Kramer;

<Номер выхода> - номер выхода устройства Kramer.

Для получения состояния выхода по запросу, программа должна отправить устройству modVS команду в формате:

IN?=<Номер выхода>

Где, <Номер выхода> - номер выхода устройства Kramer;

Команды управления.

Для управлений устройством предусмотрено три типа команд.

IN=<Номер входа>;<Номер выхода> - переключить аудио и видео (совместная коммутация) сигнал входа <Номер входа> на выход <Номер выхода>

Если <Номер входа> равен 0, а <Номер выхода> не равен нулю, то выход <Номер выхода> выключается (прекращается коммутация любого входа на этот выход).

Если <Номер входа> равен 0 и <Номер выхода> равен нулю, то все выходы выключаются (прекращается коммутация любого входа на любой выход).

INA=<Номер входа>;<Номер выхода> - переключить аудио сигнал входа <Номер входа> на выход <Номер выхода>

INV=<Номер входа>;<Номер выхода> - переключить видео сигнал входа <Номер входа> на выход <Номер выхода>

Управление осуществляется командой SEND_COMMAND на виртуальное устройство vdVS.

Например:

SEND_COMMAND vdVS,""IN=3;5"" – переключить вход номер 3 на выход номер 5.

SEND_COMMAND vdVS,""IN?=5"" – узнать какой вход переключен на выход номер 5.

Подключение модуля к программе AMX.

Для подключения модуля к программа AMX, необходимо:

1. В секции **DEFINE_DEVICE**

определить порт, к которому подключено устройство Kramer и задать виртуальное устройство для связи с модулем modVSI:

DEFINE_DEVICE

```
dvVA = 5001:1:0 // 232/485 порт на процессоре AMX, к которому подключено устройство Kramer
vdvVA16 = 33001:1:0 // Виртуальное устройство для связи с модулем modVA16xl
```

2. В секции **DEFINE_START**

описать модуль для связи с устройством Kramer:

DEFINE_START

```
DEFINE_MODULE 'modVS' modKramerVS(dvVS, vdvVS) // Модуль работы с устройством Kramer
```

Модуль CAM установит параметры порта RS232 для работы с оборудованием Kramer.

Сообщения об ошибках

При получении неправильной команды из программы AMX, модуль выдает сообщение следующего вида:

```
"VS_ERROR. UNNONE COMMAND =<Неправильная команда>"
```