



Украина, г. Киев, ул. Лебедева-Кумача 6, оф. 48;  
тел: (+38067) 442-33-89, (+38044) 496-29-60  
E-mail: [sales@akon.com.ua](mailto:sales@akon.com.ua) Internet: <http://www.akon.com.ua>

# РАСПРЕДЕЛЁННАЯ ШИНА ISA

## Концепция построения и руководство по программированию

(API Isabox.dll)

Киев – 2006

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1. КОНЦЕПЦИЯ ПОСТРОЕНИЯ ШИНЫ .....	3
2. ФУНКЦИИ ISABOX.DLL .....	5
2.1. ФУНКЦИИ РЕГИСТРАЦИИ И СБОРКИ .....	6
2.2. СЕРВИСНЫЕ ФУНКЦИИ .....	7
2.3. ФУНКЦИИ ДОСТУПА К ПОРТАМ ШИНЫ ISA .....	8
3. ПРИМЕР СБОРКИ ШИНЫ И ДОСТУПА К ПОРТАМ .....	10

## Введение

С развитием вычислительной техники и постоянного роста потребностей в высокоскоростных устройствах производители персональных компьютеров отказываются от таких периферийных узлов как шина ISA и последовательных портов RS-232, LPT. По этой причине приходится снимать с производства проверенное временем не дорогое, но надежное оборудование, использующее эти интерфейсы. Стоимость PCI –плат по сравнению с платами на ISA выше в несколько раз, но проблема заключается ещё и в том, что не всегда нужна высокая производительность PCI –плат. Компания «АКОН» предлагает альтернативный вариант шины ISA, построенной на контроллерах ISA-BOX. Это **внешняя** распределенная шина, которая состоит из одного или нескольких контроллеров в зависимости от задач, которые управляют подключенными к ним платами ISA. Программное обеспечение верхнего уровня представляет набор внешних контроллеров как единое целое, т.е. как одну шину. Собранная таким образом внешняя шина совместима с платами всех производителей. К особенностям такого решения можно отнести отсутствие отработки прерываний и каналов прямого доступа к памяти. К преимуществам то, что шина собрана на внешних модулях (при этом не нужно открывать корпус компьютера) и то, что её можно собирать даже в том случае, если на компьютере есть слоты ISA, но в недостаточном количестве.

### 1. Концепция построения шины

Узловые контроллеры ISA-BOX могут подключаться к таким интерфейсам как RS-232, RS-485, USB, Ethernet (витая пара).

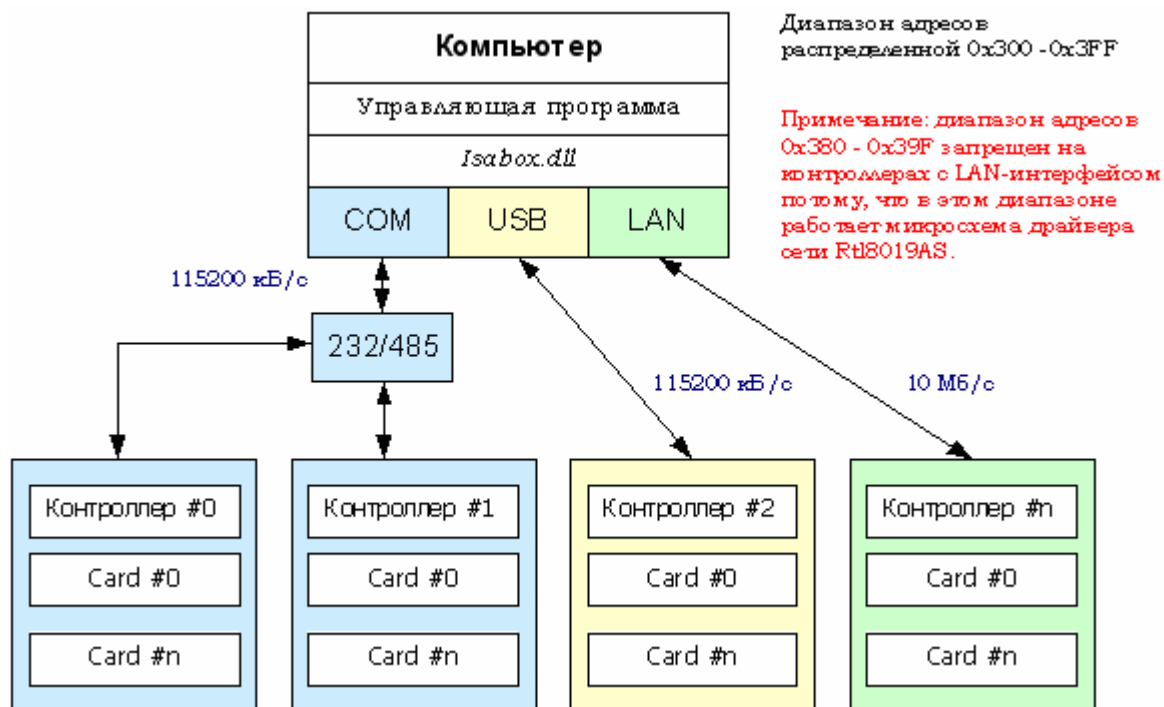


Рис.1 Пример построения распределенной шины

Управление контроллерами берет на себя библиотека *Isabox.dll*. Порядок настройки шины заключается в том, что в библиотеке регистрируются интерфейсы, к которым подключены контроллеры и сами контроллеры. Если контроллер подключен к интерфейсу RS-232, RS-485 или USB, то его адреса лежат в диапазоне от 1 до 255. Для контроллеров подключенных к LAN назначаются IP-адреса. Если адрес контроллера не известен, то его можно узнать с помощью программы *Isabox Administrator*, вызвав пункт меню «Поиск». (См. описание на программу *IsaboxAdm*). При этом к указанному интерфейсу должен быть подключен ТОЛЬКО ОДИН контроллер, потому что в этом режиме посылаются широковещательные запросы. Для RS/USB это адрес *0x00*, а для LAN *0xFFFFFFFF*.

Из прикладной программы перед началом сборки шины нужно вызвать функции *ibcClearInterfaceList* и *ibcClearControllerList* для очистки реестра интерфейсов и контроллеров. Для регистрации интерфейсов вызываются функции *ibcOpenInterface* и *ibcOpenLanInterface*. Первая функция вызывается при регистрации COM-порта. Контроллеры с USB интерфейсом создают при подключении виртуальный COM-порт, поэтому эта функция вызывается по тем же правилам. Вторая функция вызывается в том случае, если есть контроллеры, подключенные к LAN. Обе функции при успешном выполнении возвращают идентификатор интерфейса. Этот идентификатор используется при регистрации контроллеров. Для регистрации контроллеров вызывается функция *ibcOpenController*. При регистрации контроллера нужно указать идентификатор интерфейса, к которому подключен контроллер, адрес контроллера и диапазон адресов шины ISA, которые он будет обслуживать. При регистрации контроллеров адреса шины ISA нужно распределять так чтобы они не перекрывались с адресами, зарегистрированными на других контроллерах, и не выходили за диапазон *0x300..0x3FF*. Если у контроллера LAN интерфейс, то в этом случае запрещается использовать на нем диапазон адресов *0x380..0x39F*, потому что в этом диапазоне работает микросхема драйвера сети *Rtl8019AS*. Если есть контроллеры, подключенные к LAN, то нужно вызвать функцию *ibcSetLocalPort* чтобы указать номер порта на компьютере. После этих манипуляций вызывается функция *ibcControl*, которая производит подключение к COM-портам и открывает UDP-сокеты. Теперь шина ISA собрана и готова к использованию.

## 2. Функции Isabox.dll

Весь набор функций библиотеки Isabox.dll можно разделить на три группы:

№	Группы	Функции
1	функции регистрации и сборки	ibcSetLocalPort ibcGetLocalPort ibcOpenInterface ibcOpenLanInterface ibcOpenController ibcClearControllerList ibcClearInterfaceList ibcControl
2	Сервисные функции	ibcGetControllerProfile ibcGetObjectList ibcFirstObject ibcNextObject ibcEofObject ibcGetOncObject ibcSetRsCfg ibcGetRsCfg ibcSetLanCfg ibcGetLanCfg ibcSaveToFlash ibcRestoreFromFlash
3	Функции доступа к портам шины ISA	ibcReadPort ibcWritePort ibcReadPortW ibcWritePortW ibcExReadPort ibcExWritePort ibcExReadPortW ibcExWritePortW

## 2.1. Функции регистрации и сборки

*ibcClearInterfaceList*  
*ibcClearControllerList*

**Очистка реестра интерфейсов и контроллеров**

```
void ibcClearControllerList(void);
void ibcClearInterfaceList(void);
```

**Параметры:** отсутствуют;

*ibcOpenInterface*

**Регистрация COM-порта**

```
unsigned long ibcOpenInterface(unsigned char ComPort,
                               unsigned long Baudrate,
                               HANDLE OpenPortHandle);
```

**Параметры:**

1. ComPort - номер COM-порта; (0 - COM1, 1 - COM2 и т.д.);
2. Baudrate - скорость обмена; (1200 - 115200);
3. OpenPortHandle - если есть необходимость подключить к одному порту различные устройства (на RS-485), то этот порт нужно открыть до его регистрации в библиотеке и хендл открытого порта передать третьим параметром. Если такой надобности нет, то приравнять его к NULL.

**Возвращаемый параметр:** идентификатор интерфейса. Используется при регистрации контроллеров. (См. ниже)

*ibcOpenLanInterface*

**Регистрация LAN интерфейса**

```
unsigned long ibcOpenLanInterface(void);
```

**Параметры:** отсутствуют

**Возвращаемый параметр:** идентификатор интерфейса. Используется при регистрации контроллеров. (См. ниже)

***ibcOpenController*****Регистрация контроллера**

```
unsigned long ibcOpenController(unsigned long InterfaceHandle,  
                               unsigned long Address,  
                               unsigned short StartAddress,  
                               unsigned short StopAddress);
```

**Параметры:**

1. InterfaceHandle - идентификатор интерфейса возвращаемый функциями регистрации интерфейсов;
2. Address - адрес контроллера; ( 1 - 255 для RS, 0 - широковещательный) пример для LAN = 0xC0A80102 = 192.168.1.2;
3. StartAddress, StopAddress - диапазон адресов шины ISA которые будет обслуживать контроллер.

**Возвращаемый параметр:** идентификатор контроллера. Используется при вызове сервисных функций для получения дополнительной информации о контроллере. (См. ниже)

***ibControl*****Включение/выключение шины**

```
bool ibControl(bool status);
```

**Параметры:**

1. status - (true/false);

**Возвращаемый параметр:** в случае успешного подключения к зарегистрированным COM-портам и создания сокетов функция возвращает **true**. В противном случае - **false**.

## 2.2. Сервисные функции

Сервисные функции не нужны для использования в прикладной программе для работы с шиной ISA. Если все же они требуются, то их детальное описание предоставляется компанией «АКОН» в отдельном документе.

## 2.3. Функции доступа к портам шины ISA

При вызове функций доступа к портам шины ISA библиотека сама выбирает, какой контроллер будет обрабатывать запрос. Это возможно благодаря тому что на этапе регистрации контроллеров были указаны диапазоны адресов шины ISA для каждого узлового контроллера.

*ibcReadPort*  
*ibcWritePort*  
*ibcReadPortW*  
*ibcWritePortW*

**Побайтный и пословный доступ к одному порту**

```
bool ibcReadPort(unsigned short address, unsigned char *value);
bool ibcWritePort(unsigned short address, unsigned char value);
bool ibcReadPortW(unsigned short address, unsigned short *value);
bool ibcWritePortW(unsigned short address, unsigned short value);
```

### Параметры:

1. address - адрес порта шины ISA;
2. value - значение, записываемое в порт или считываемое с порта.

**Возвращаемый параметр:** в случае успешного выполнения функция возвращает `true`. В противном случае - `false`. Причиной не успешного вызова может стать либо не правильно зарегистрированный интерфейс и/или контроллер либо плохая линия связи с контроллером.

*ibcExReadPort*  
*ibcExWritePort*  
*ibcExReadPortW*  
*ibcExWritePortW*

**Побайтный и пословный доступ к одному порту в режиме записи/чтения массива данных.**

```
bool ibcExReadPort(unsigned short address,  
                  unsigned char *data,  
                  unsigned short size);  
  
bool ibcExWritePort(unsigned short address,  
                   unsigned char *data,  
                   unsigned short size);  
  
bool ibcExReadPortW(unsigned short address,  
                   unsigned short *data,  
                   unsigned short size);  
  
bool ibcExWritePortW(unsigned short address,  
                    unsigned short *data,  
                    unsigned short size);
```

**Параметры:**

1. address - адрес порта шины ISA;
2. data - указатель на массив, записываемых в порт или считываемых с порта значений.
3. size - количество элементов массива

**Возвращаемый параметр:** в случае успешного выполнения функция возвращает **true**. В противном случае - **false**. Причиной не успешного вызова может стать либо не правильно зарегистрированный интерфейс и/или контроллер либо плохая линия связи с контроллером.

### 3. Пример сборки шины и доступа к портам

В данном примере подключим два устройства: первое на COM2 и второе к LAN. Скорость обмена по COM-порту 115200 кбит/с, адрес контроллера 0x05. IP контроллера подключенного к LAN равен 192.168.1.6. Диапазон адресов ввода/вывода обслуживаемый первым контроллером равен: 0x300..0x309 включительно. Диапазон адресов ввода/вывода обслуживаемый вторым контроллером: 0x310..0x319 включительно. Номер порта для UDP-сокета назначим, к примеру, равным 5100. После сборки шины произведем чтение с порта 0x300 и запись в порт 0x310.

```
#define COM2 1
#define LOCAL_PORT 5100
void main(void)
{
    unsigned long ComInterface, ComController;
    unsigned long LanInterface, LanController;
    unsigned short PortValue;
    // 1. Очистка реестров интерфейсов и контроллеров
    ibcClearControllerList();
    ibcClearInterfaceList();
    // 2. Регистрация интерфейсов
    ComInterface = ibcOpenInterface(COM2, 115200, NULL);
    LanInterface = ibcOpenLanInterface();
    // 3. Регистрация контроллеров
    ComController = ibcOpenController(ComInterface,
                                     0x05,
                                     0x300,
                                     0x309);
    LanController = ibcOpenController(LanInterface,
                                     0xC0A80106,
                                     0x310,
                                     0x319);

    // 4. Номер UDP-порта
    ibcSetLocalPort(LOCAL_PORT);
    // 5. Сборка шины
    ibcControl(true);
    // 6. Доступ к портам
    ibcReadPortW(0x300, &PortValue); // Чтение с порта
    ibcWritePortW(0x310, PortValue); // Запись в порт
    // 7. Демонтаж шины ISA
    ibcControl(false);
}
```

Модуль разработан и изготовлен Компанией АКОН.  
Предлагаем к поставке модули АЦП, модули ЦАП,  
устройства ввода-вывода цифровой информации,  
модули нормирующих преобразователей с гальванической развязкой, модули  
для распределённых систем и другое оборудование.

Украина, г. Киев,  
ул. Лебедева-Кумача 6, 48  
тел.(8067) 442-33-89, (8044) 496-29-60  
E-mail: [sales@akon.com.ua](mailto:sales@akon.com.ua)  
Http: [www.akon.com.ua](http://www.akon.com.ua)