

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
кафедра молекулярной физики

ИНСТРУКЦИЯ ПО ЭКСПЛУАТАЦИИ
**ДРАЙВЕР МАСС-СПЕКТРОМЕТРА *МИ 1201-АГМ*
ДЛЯ WINDOWS NT 4.0**

(приложение к заключительному отчету по договору)

Х.Д. № 25/16/1427

Разработчик
электроник 11 разряда

__._.01

Климов Е.А.

Екатеринбург 2001

РЕФЕРАТ

Климов Е.А. *ДРАЙВЕР МАСС-СПЕКТРОМЕТРА МИ 1201-АГМ ДЛЯ WINDOWS NT 4.0*: Инструкция по эксплуатации. Екатеринбург: УГТУ, 2001. 13 с.

Рис. 3. Табл. 6.

ДРАЙВЕР; WINDOWS NT; МАСС-СПЕКТРОМЕТР; МИ 1201-АГМ.

Описаны принципы функционирования драйвера режима ядра ОС Windows NT, предназначенного для обеспечения доступа к портам ввода-вывода МИ 1201-АГМ через файловую систему ОС.

Описан драйвер виртуальной машины DOS, предназначенный для функционирования стандартной программы управления масс-спектрометром МИ 1201-АГМ, разработанной для платформы DOS.

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ СИМВОЛОВ, ЕДИНИЦ И ТЕРМИНОВ	4
ВВЕДЕНИЕ	4
1. ДРАЙВЕР РЕЖИМА ЯДРА ОС ДЛЯ МИ 1201-АГМ	5
1.1. Принципы работы Windows NT с внешними устройствами	5
1.2. Общая структура KMD	5
1.3. Процесс инициализации «PortIO.sys»	6
1.4. Обработка запросов «PortIO.sys»	6
1.5. Процесс выгрузки драйвера «PortIO.sys»	7
1.6. Работа с драйвером «PortIO.sys» через вызовы функций ОС	7
1.6.1. CreateFile()	8
1.6.2. CloseHandle()	8
1.6.3. DeviceIOControl()	9
1.7. Установка драйвера «PortIO.sys»	9
2. ВИРТУАЛЬНОЕ УСТРОЙСТВО DOS	11
2.1. Принципы работы виртуального устройства DOS	11
2.2. Общая структура виртуального драйвера DOS	11
2.3. Процесс инициализации «MI1201.DLL»	12
2.4. Обработка запросов к портам ввода-вывода	12
2.5. Процесс выгрузки драйвера	13
2.6. Установка драйвера	13
ЗАКЛЮЧЕНИЕ	13

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ СИМВОЛОВ, ЕДИНИЦ И ТЕРМИНОВ

- VDD – Virtual Dos Device, виртуальное устройство DOS.
DLL – Dynamic Link Library, динамически подключаемая библиотека
KMD – Kernel Mode Driver, драйвер режима ядра.

ВВЕДЕНИЕ

Управление масс-спектрометром МИ1201-АГМ требует обращения к портам ввода-вывода процессора¹⁾. Операционная система Windows NT 4.0 блокирует непосредственную работу прикладных программ с портами ввода-вывода процессора. Для поддержки работы прикладных программ с портами ввода-вывода процессора необходим драйвер устройства, работающий в режиме ядра системы (kernel mode driver или KMD). Для работы DOS-программ, обращающихся непосредственно к портам ввода-вывода процессора, также необходим драйвер предназначенный для DOS-окна Windows NT и преобразующий эти обращения в обращения к драйверу ядра (virtual DOS device, VDD).

В настоящем документе описаны процедура установки и работа с драйвером ядра и виртуальным драйвером DOS предназначенными для управления масс-спектрометром МИ1201-АГМ.

Кратко описаны также базовые принципы работы с внешними устройствами в операционной системе Windows NT 4.0.

¹⁾ Порты ввода-вывода процессора применяются для управления внешними устройствами.

1. ДРАЙВЕР РЕЖИМА ЯДРА ОС ДЛЯ МИ 1201-АГМ

1.1. ПРИНЦИПЫ РАБОТЫ WINDOWS NT С ВНЕШНИМИ УСТРОЙСТВАМИ

Для управления масс-спектрометром под Windows NT необходим доступ к портам ввода-вывода процессора. Прямой доступ к портам ввода-вывода блокируется ОС Windows NT. Доступ к портам возможен только через специальный драйвер Windows NT.

В самом общем и очень упрощенном виде схема взаимодействия прикладных программ Windows NT с портами ввода-вывода¹⁾ приведена на рис. 1.1.

Для управления масс-спектрометром МИ1201-АГМ создан драйвер ядра «PortIO.sys», описываемый ниже.

1.2. ОБЩАЯ СТРУКТУРА KMD

KMD «PortIO.sys» — программа на языке Си. Исходный текст программы находится см. в папке «NT-драйверы для МИ 1201-АГМ\Исходный код драйверов\Драйвер ядра NT\».

Драйвер является переработанной версией стандартного драйвера ввода-вывода в порты, поставляемого Microsoft Corp. в составе «Microsoft® Windows NT® Version 4.0 Device Driver Kit». В результате модификации драйвер поддерживает несколько диапазонов портов. Перечень функций «PortIO.sys» приведен в табл. 1.1.

ВЗАИМОДЕЙСТВИЕ ПРИКЛАДНЫХ ПРОГРАММ WINDOWS NT С ПОРТАМИ ВВОДА-ВЫВОДА

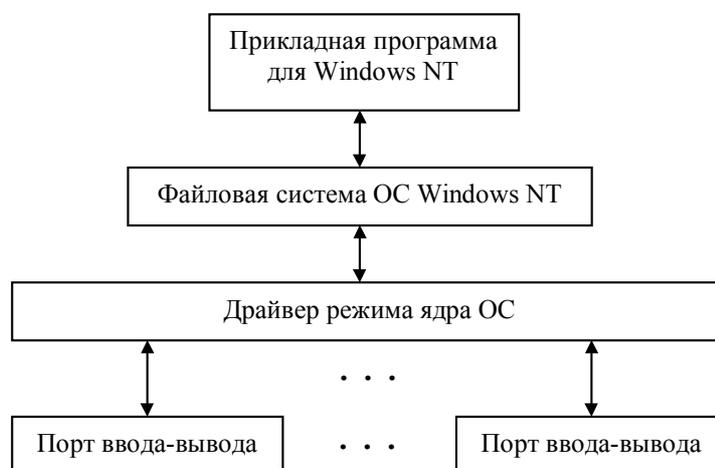


Рис. 1.1

Таблица 1.1

ФУНКЦИИ «PORTIO.SYS»

Функция	Назначение
1. DriverEntry()	Инициализация, чтение параметров из реестра, создание устройства вызовом GpdCreateDevice()
2. GpdCreateDevice()	Создание устройства
3. GpdDispatch()	Обработка запросов (чтение, запись, выгрузка) вызовом соответствующих функций (GpdioctlReadPort(), GpdioctlWritePort(), GpdUnload())
4. GpdioctlReadPort()	Чтение данных из порта
5. GpdioctlWritePort()	Запись данных в порт
6. GpdUnload()	Выгрузка

1.3. ПРОЦЕСС ИНИЦИАЛИЗАЦИИ «PORTIO.SYS»

- 1) ОС вызывает функцию драйвера DriverEntry() (эта функция присутствует в любых KMD).
- 2) DriverEntry() считывает из реестра ключ
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PortIO\Parameters
параметры устройства, т. е., в данном случае, диапазоны используемых портов ввода-вывода. Задание диапазонов портов осуществляется с помощью значений вида «IoPortAddressN» и «IoPortCountN», где *N* может принимать значения от 0 до 9.
- 3) DriverEntry() сообщает ОС о предполагаемом использовании портов с целью исключить возможные конфликты между существующими драйверами.
- 4) В случае отсутствия конфликтов DriverEntry() создает устройство и передает ОС адрес функции-диспетчера запросов (GpdDispatch()).

1.4. ОБРАБОТКА ЗАПРОСОВ «PORTIO.SYS»

Запросы, обрабатываемые функцией GpdDispatch() «PortIO.sys», перечислены в табл. 1.2.

ЗАПРОСЫ НА ВВОД-ВЫВОД ДЛЯ «PORTIO.SYS»

Запрос	Условие вызова	Реакция
1. IRP_MJ_CREATE	Вызывается ОС во время инициализации драйвера	Сообщение ОС об успешном запуске драйвера
2. IRP_MJ_CLOSE	Вызывается ОС во время выгрузки драйвера	Сообщение ОС об успешной выгрузке драйвера
3. IRP_MJ_DEVICE_CONTROL	Вызывается при использовании драйвера для чтения/записи	Вызов GpdiocReadPort()/GpdiocWritePort()

1.5. ПРОЦЕСС ВЫГРУЗКИ ДРАЙВЕРА «PORTIO.SYS»

Для выгрузки ОС вызывает функцию GpdUnload(), которая выполняет следующие действия.

- 1) Сообщает ОС о том, что драйвер более не использует порты.
- 2) Разрушает устройство.

1.6. РАБОТА С ДРАЙВЕРОМ «PORTIO.SYS» ЧЕРЕЗ ВЫЗОВЫ ФУНКЦИЙ ОС

Взаимодействие прикладной программы с «PortIO.sys» осуществляется через вызовы системных функций создания, уничтожения и передачи данных, приведенные в табл. 1.3.

Таблица 1.3

СИСТЕМНЫЕ ФУНКЦИИ ДЛЯ РАБОТЫ С KMD

Функция	Назначение
1. CreateFile()	Создает файловое устройство, представляющее диапазоны портов.
2. CloseHandle()	Уничтожает дескриптор, созданный предыдущей функцией.
3. DeviceIOControl()	Работа с файловым устройством - чтение/запись.

1.6.1. CreateFile()

Для работы с портами необходимо создать файловое устройство посредством вызова `CreateFile()`. В качестве параметров передается имя файлового устройства, созданного «PortIO.sys», режим доступа (запись/чтение), режим совместного доступа, параметры защищенности, разрешение на создание файла, атрибуты и шаблон. Большинство этих параметров для работы несущественны, типичные значения параметров приведены в табл. 1.4. Пример подключения к драйверу

```
HDevice := CreateFile(
    '\\.\PortIODev',    // имя драйвера
    GENERIC_READ or GENERIC_WRITE, // тип доступа
    FILE_SHARE_READ,  // разрешение совместного доступа
    nil,               // адрес описания прав доступа
    OPEN_EXISTING,    // как создавать файл
    0,                // атрибуты файла
    0                 // идентификатор файла для копирования атрибутов файла
);
```

В переменной *HDevice* будет возвращен идентификатор файла (handle) для доступа к нему.

1.6.2. CloseHandle()

Для завершения работы с портами необходимо уничтожить файловое устройство посредством вызова `CloseHandle()`. В качестве параметра передается дескриптор файлового устройства, созданного ранее вызовом `CreateFile()`. Например:

Таблица 1.4

ПАРАМЕТРЫ, ПЕРЕДАВАЕМЫЕ ФУНКЦИИ `CREATEFILE()`²

Тип	Значение
1. строка	\\.\PortIODev
2. двойное слово	GENERIC_READ or GENERIC_WRITE
3. двойное слово	FILE_SHARE_READ
4. указатель	nil
5. двойное слово	OPEN_EXISTING
6. двойное слово	0
7. дескриптор	0

² Все параметры здесь и далее приведены в синтаксисе Delphi.

`CloseHandle(HDevice);`

данный вызов освобождает системные ресурсы и прекращает доступ к файлу.

1.6.3. DeviceIOControl()

Работа с портами происходит посредством вызовов функции работы с файловыми устройствами `DeviceIOControl()`. В качестве параметров передается дескриптор устройства, управляющий код, буферы и их размеры для входной и выходной информации, а также длина возвращенной информации и указатель на структуру для асинхронной работы с устройством. Управляющий код формируется из битовых полей, говорящих системе о функциональности драйвера и вызываемой функции. Например, запись в порт:

```
DeviceIOControl(
    HDevice,      // идентификатор файла
    IOCTL_GPD_WRITE_PORT_UCHAR, // код операции
    @buf,        // указатель на буфер с данными вывода
    SizeOf(buf), // размер буфера с данными вывода
    nil,         // указатель на буфер для получения данных ввода
    0,          // размер буфера для получения данных
    ret,        // переменная для размера возвращаемых данных
    nil        // указатель на структуру асинхронного ввода-вывода
);
```

аналогично чтение порта:

```
DeviceIOControl(
    HDevice,      // идентификатор файла
    IOCTL_GPD_READ_PORT_UCHAR, // код операции
    @TmpPort,    // указатель на буфер с данными вывода
    SizeOf(TmpPort), // размер буфера с данными вывода
    @Result,     // указатель на буфер для получения данных ввода
    1,          // размер буфера для получения данных
    ret,        // переменная для размера возвращаемых данных
    nil        // указатель на структуру асинхронного ввода-вывода
);
```

1.7. УСТАНОВКА ДРАЙВЕРА «PORTIO.SYS»

- 1) Драйвер (файл «PortIO.sys») должен быть помещен в папку
System32\Drivers

главной папки системы (обычно это C:\WinNT, т.е. полный путь будет выглядеть как C:\WinNT\System32\Drivers).

- 2) В системном реестре в ветви

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\

должен быть создан ключ с именем, совпадающим с именем файла драйвера без расширения:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PortIO

Значения, которые необходимо создать в данном ключе, перечислены в табл. 1.5. Данный ключ дает информацию о драйвере системе.

- 3) В указанном ключе создается ключ Parameters, который служит для указания параметров драйверу. В данном случае здесь перечисляются диапазоны портов ввода-вывода. Значения типа двойное слово, записанные в этот ключ должны быть названы IoPortAddress* и IoPortCount*, где «*» пробегает значения от 0 до 9. Например, если необходимы два диапазона портов, то записываются значения IoPortAddress0, IoPortCount0, IoPortAddress1, IoPortCount1. IoPortAddress отвечает базовому порту (порту с наименьшим адресом), а IoPortCount - количеству последовательных портов.

Необходимые установки записаны в файле «PortIO.reg», их можно переписать в системный реестр дважды щелкнув мышкой на указанном файле.

Установки файла «PortIO.reg» предписывают для драйвера старт

Таблица 1.5

ПАРАМЕТРЫ РЕЕСТРА ДЛЯ «PORTIO.SYS»

Название значения	Тип значения	Значение	Комментарии
1. Type	Двойное слово	1	
2. Start	Двойное слово	3	Старт производится вручную («Manual») см. Control Panel→Devices, после успешного запуска можно изменить значение на «Automatic».
3. Group	Строка	Extended Base	
4. ErrorControl	Двойное слово	1	
5. DisplayName	Строка	МИ 1201 port I/O	Определяет название драйвера, которое будет отображаться в Devices(Устройства) панели управления

«вручную» («Manual»), см. Control Panel→ Devices. Это сделано с целью обеспечить безопасность системы, если при запуске драйвер приведет к остановке ОС, то можно просто загрузить ОС заново. После успешного запуска можно изменить значение на «Automatic» — драйвер будет запускаться при каждом старте ОС.

2. ВИРТУАЛЬНОЕ УСТРОЙСТВО DOS

2.1. ПРИНЦИПЫ РАБОТЫ ВИРТУАЛЬНОГО УСТРОЙСТВА DOS

Для работы уже существующей программы для ОС MS-DOS под Windows NT 4.0 необходимо виртуальное устройство DOS (драйвер виртуальной машины DOS), которое получает управление при перехвате ОС обращения к портам ввода-вывода и заменяет их на вызовы функций «PortIO.sys».

Схема взаимодействия прикладных программ DOS с портами ввода-вывода в DOS приведена на рис. 2.1 и в ОС Windows NT приведена на рис. 2.2.

2.2. ОБЩАЯ СТРУКТУРА ВИРТУАЛЬНОГО ДРАЙВЕРА DOS

Виртуальный драйвер DOS «MI1201.DLL» является подключаемой библиотекой, написанной на языке Си. Исходный текст программы см. в папке «NT-драйверы для МИ 1201-АГМ\Исходный код драйверов\Виртуальный драйвер DOS (VDD)\».

Перечень функций «MI1201.DLL» приведен в табл. 2.1.

ВЗАИМОДЕЙСТВИЕ ПРИКЛАДНЫХ ПРОГРАММ DOS С ПОРТАМИ ВВОДА-ВЫВОДА В РЕАЛЬНОМ РЕЖИМЕ

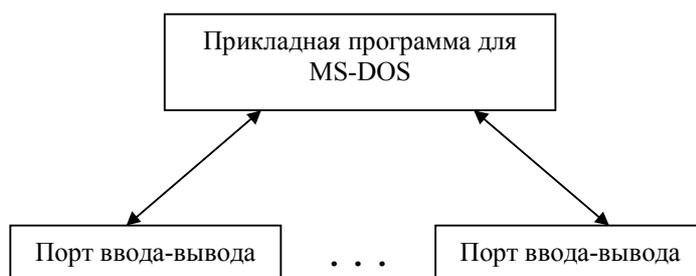


Рис. 2.1

ВЗАИМОДЕЙСТВИЕ ПРИКЛАДНЫХ ПРОГРАММ DOS С ПОРТАМИ ВВОДА-ВЫВОДА В СРЕДЕ WINDOWS NT

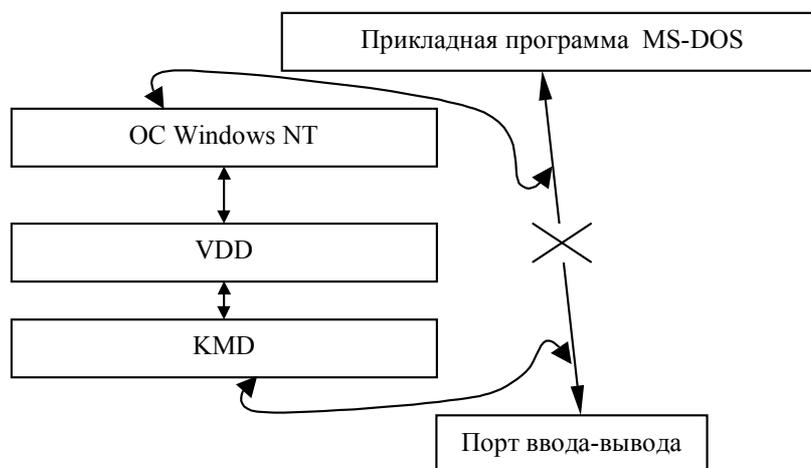


Рис. 2.2

2.3. ПРОЦЕСС ИНИЦИАЛИЗАЦИИ «MI1201.DLL»

При инициализации «MI1201.DLL» сообщает системе, что при попытке прямого использования портов из указанных VDD диапазонов программой DOS управление необходимо передать функции VDD, которая при инициализации объявляется как обработчик таких запросов.

2.4. ОБРАБОТКА ЗАПРОСОВ К ПОРТАМ ВВОДА-ВЫВОДА

Все попытки программы DOS обратиться к портам перехватываются ОС и она производит поиск запущенного VDD, диапазоны портов которого включают в себя данный порт. Если система находит такой VDD, то она передает управление функции, которая при инициализации VDD была объявлена как об-

Таблица 2.1

ФУНКЦИИ «MI1201.DLL»

Функция	Назначение
1. VDDInitialize()	Инициализация / освобождение.
2. MyInB()	Эмуляция чтения байта из порта. На самом деле происходит обращение к KMD.
3. MyOutB()	Эмуляция записи байта в порт. На самом деле происходит обращение к KMD.

работчик подобных запросов. В описываемом «MI1201.DLL» такими функциями являются MyInB() и MyOutB(), отвечающие соответственно за ввод и вывод из/в порты. Данные функции обращаются к драйверу ядра «PortIO.sys». С точки зрения программы DOS подобные цепочки обращений абсолютно прозрачны, т.е. программа не может отличить реальное обращение к портам от их перехвата и передачи управления цепочке драйверов.

2.5. ПРОЦЕСС ВЫГРУЗКИ ДРАЙВЕРА

При выгрузке «MI1201.DLL» просто сообщает ОС то, что ОС не должна более передавать «MI1201.DLL» запросы об обращении к портам и завершает работу.

2.6. УСТАНОВКА ДРАЙВЕРА

В реестре в ключе

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\VirtualDeviceDrivers находится значение с именем «VDD» мультистрокового типа. В это значение необходимо добавить строку, содержащую полное имя драйвера (например, c:\winnt\system32\mi1201.dll). После этого ОС будет запускать «MI1201.DLL» при каждом старте виртуальной машины DOS.

Для правильной работы «MI1201.DLL» должен быть установлен драйвер ядра «PortIO.sys».

ЗАКЛЮЧЕНИЕ

Описанные выше драйверы позволяют работать с масс-спектрометром МИ 1201-АГМ в операционной системе Windows NT 4.0. Эта операционная система является наиболее устойчивой и защищенной из операционных систем персональных ЭВМ.

Драйверы позволяют получить доступ ко всем преимуществам этой ОС, и допускают одновременное использование унаследованного ПО для MS-DOS.

Как важное достоинство работы в ОС Windows NT можно указать возможность разграничения доступа — т.е. есть возможность доработать драйвер ядра и явно назначать права на доступ к оборудованию определенным пользователям.