

# Драйвер связи с модулями ввода-вывода контроллеров серии Fimatic-C

FimaticDRV V2.1

Описание. Общая информация

Разработал :

Клабуков М.Ю.

ООО “БПА” ИЦ. Москва 2023г

# 1 Введение

В документе приводится краткое описание драйвера FimaticDRV. Драйвер разработан для модуля МП300 (модуля процессорного, серии Fimatic-C, компании “Физприбор”). Драйвер предназначен для организации обмена данными между модулями ввода-вывода серии Fimatic и прикладными программами пользователя, функционирующими на модуле МП300 под управлением операционной системы Linux.

Драйвер относится к типу драйверов последовательного интерфейса UART (Universal Asynchronous Receiver-Transmitter). На физическом уровне обмен с модулями ввода-вывода осуществляется через полудуплексную шину RS485. При этом одной из задач драйвера является управление (переключение) между режимами прием-передача на шине RS485.

## 2 Возможности драйвера

- настройка пользователем скорости обмена (57600/ 115200 bps) при запуске драйвера;
- регистрация драйвера в ядре Linux, конфигурация аппаратных ресурсов UART 16C750 согласно Device Tree файла, настройка подсистем (питания, синхронизации, прерывания);
- получение от программы пользователя (через собственный API интерфейс на основе IOCTL) запросов на обмен с модулями IO;
- в зависимости от вида запроса пользователя, для каждого типа модуля, в соответствие с протоколом Fimatic-Bus, формирование пакета запроса к модулю IO и его отправка;
- использование блокирующих системных вызовов (wait\_event) для уменьшения нагрузки на систему на период ожидания ответа;
- получение от модулей IO ответных пакетов и их обработка в режиме прерываний. Вычисление контрольных сумм, контроль достоверности и целостности и посредством API интерфейса предоставление информации программе пользователя;
- диагностика следующих видов ошибок: «превышено время ожидания ответа» (TimeOut), «Ошибка контрольной суммы» (CRC), «Ошибка в ответе модуля». Выявленный код ошибки через API интерфейс возвращается программе пользователя;
- создание и обработка диагностических счетчиков. Счетчики реализованы для сетевых узлов с адресами: 1...31. Для каждого модуля, установленного в этом диапазоне, с нарастающим итогом ведется накопление, отдельно по каждому из трех видов ошибок. Счетчики могут быть прочитаны и сброшены через API интерфейс приложением пользователя. Также существует команда группового, одновременного сброса всех счетчиков драйвера;
- в состав пакета драйвера включены исходные коды, а также откомпилированные версии тестовых утилит. Программы позволяют: протестировать драйвер, изучить API интерфейс. Также данные утилиты можно использовать для тестирования и диагностики модулей ввода/ вывода семейства Fimatic;

## 3 Установка и настройка драйвера

### 3.1 Системные требования

Драйвер выполнен как загружаемый модуль ядра Linux. Он функционирует в пространстве ядра (Kernel Space) в виде обработчиков прерываний и потоков ядра в привилегированном режиме. Разработан как платформенный драйвер и соответствует стандартной для последних версий Linux модели LDM (Linux Device Model). Поскольку драйвер регистрирует Platform Driver, для связывания устройства и драйвера платформы, используется ключевое слово «fimatic». Оно определено как в структуре «my\_of\_ids» драйвера, так и должно быть описано в теге «compatible» файла Device Tree:

```
466 // Structure: "my_of_ids" callback -----
467 //
468 static const struct of_device_id my_of_ids[
469     { .compatible = "fimatic" },
470     {},
471 ];
472
```

```
34 &uart1 {
35     compatible = "fimatic";
36     status = "okay";
37     pinctrl-names = "default";
38     pinctrl-0 = <uart1_pins>;
39
40 };
```

### 3.2 Работа с драйвером

Драйвер загружается как модуль ядра командой insmod:

```
root@am335x-evm:~# insmod Fimatic_Drv_Main.ko prxBaudrate=115200
[ 122.514733] FimaticDrv: Fimatic Bus Drv. V1.2 2023
[ 122.519718] FimaticDrv: set data exchange rate: 115200
[ 122.524966] FimaticDrv: init Diagnostics...
[ 122.531835] FimaticDrv: the device is compatible with: fimatic_uart ...
[ 122.538530] FimaticDrv: IRQ 35
[ 122.547567] FimaticDrv: function fimatic_uart_init. Platform driver Registered ...
```

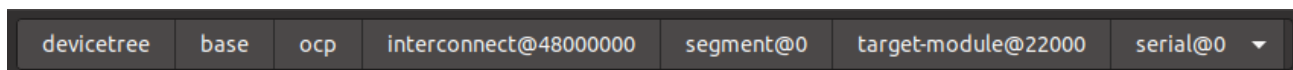
Параметр «prxBaudrate» может быть также определен для скорости «57600». Если параметр не определен, то «57600» устанавливается драйвером по умолчанию.

Чтобы убедиться в том, что драйвер загружен и система выделила для него номер прерывания, можно выполнить команды:

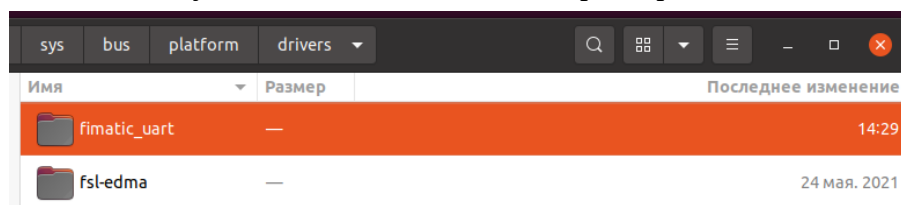
```
root@am335x-evm:~# lsmod |grep Fimatic
Fimatic_Drv_Main      20480  0
root@am335x-evm:~#
root@am335x-evm:~#
```

```
root@am335x-evm:~# cat /proc/interrupts |grep UART1_IRQ
35:                0          INTC    73 Level    UART1_IRQ
```

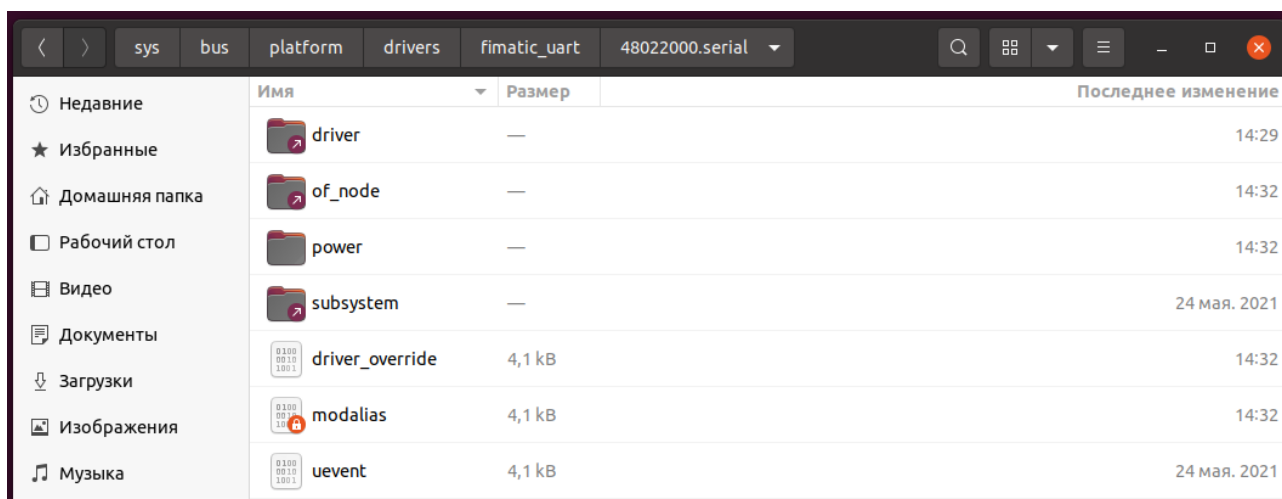
Определенная в DT конфигурация экспортируется через виртуальную файловую систему sysfs. Путь к конфигурации модуля отображается в папке «firmware» в sysfs:



В папке «drivers» sysfs можно наблюдать, что драйвер «fimatic\_uart» запущен в работу:



А также, что с данным драйвером уже связано устройство 48022000.serial, которое согласно конфигурации и соответствует порту uart с номером один:



	Имя	Размер	Последнее изменение
📁	driver	—	14:29
📁	of_node	—	14:32
📁	power	—	14:32
📁	subsystem	—	24 мая. 2021
📄	driver_override	4,1 kB	14:32
📄	modalias	4,1 kB	14:32
📄	uevent	4,1 kB	24 мая. 2021

В папке /dev должно появиться зарегистрированное драйвером устройство «fimatic\_dev». Через дескриптор файла этого устройства системный вызов IOCTL прикладного процесса пользователя будет вести обмен с драйвером:

```
root@am335x-evm:~# ls -la /dev | grep fimatic_dev
crw----- 1 root root 10, 59 Jul 25 11:29 fimatic_dev
root@am335x-evm:~#
root@am335x-evm:~#
```

```
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102
```

## 4 Описание API интерфейса драйвера

### 4.1 Общий формат

```
ioctl(my_dev, cmdIOctl_Fimatic_rdIO, &Data_IO);
```

Для работы с драйвером Linux приложение пользователя должно использовать стандартный системный вызов IOCTL. Как это показано в примере выше, для функции определяется три параметра:

#### 4.1.1 Первый параметр

Файловый дескриптор драйвера. В нашем случае это:

```
my_dev = open("/dev/fimatic_dev", 0);
if (my_dev < 0) {
    perror("Fail to open device file: /dev/fimatic_dev.");
}
```

#### 4.1.2 Второй параметр

Код (номер) команды для вызова IOCTL. Перечень возможных вызовов определен в прилагаемом к драйверу «Fimatic\_Drv.h» файле. Можно выделить две основные команды (чтение и запись):

```
#define IOC_MAGIC 'k'

#define cmdIOCtl_Fimatic_rdIO      _IOWR(IOC_MAGIC, 10, struct IOCtl_Fimatic_rdIO)
#define cmdIOCtl_Fimatic_wrIO      _IOWR(IOC_MAGIC, 12, struct IOCtl_Fimatic_wrIO)
```

и пять сервисных команд для работы с диагностическими счетчиками ошибок:

```
// Выбрать номер слота диагностики (сделать его текущим)
#define cmdIOCtl_Dgn_ChooseSlot      _IOW(IOC_MAGIC, 20, int)

// Прочитать номер текущего слота диагностики
#define cmdIOCtl_Dgn_SlotCurrent      _IOR(IOC_MAGIC, 21, int)

// Сбросить счетчики текущего слота диагностики
#define cmdIOCtl_Dgn_ResCnt          _IOW(IOC_MAGIC, 22, int)

// Сбросить счетчики всех слотов диагностики
#define cmdIOCtl_Dgn_ResAll          _IOW(IOC_MAGIC, 23, int)

// Прочитать счетчики текущего слота диагностики
#define cmdIOCtl_Dgn_RdCnt          _IOR(IOC_MAGIC, 24, struct IOCtl_Fimatic_Dgn_RdCnt)
```

#### 4.1.3 Третий параметр

Определяет переменную через которую приложение пользователя, посредством вызова IOCtl, осуществляет обмен данными с драйвером. Обычно переменная является структурой данных. Для отдельных команд данная переменная имеет тип целого, т. е. Int.

Подробно формат структур для обмена описан ниже, для каждой команды в отдельности.

### 4.2 Команда «cmdIOCtl\_Fimatic\_rdIO» (0x41)

#### 4.2.1 Общее описание

Данная команда позволяет прочитать данные с любого типа модуля IO (AI,DI,DO) из линейки модулей Fimatic.

После вызова IOCtl драйвер на основании данных структуры «IOCtl\_Fimatic\_rdIO» формирует пакет запроса к модулю, отправляет его, по прерываниям от порта принимает обратно данные или код ошибки, помещает результат в структуру «IOCtl\_Fimatic\_rdIO» и возвращает пользователю в User Space. Во время работы, посредством очереди ожидания, на время приема ответного пакета, системный вызов IOCtl переводится в состояние сна (wait\_event) а после приема полного пакета или кода ошибки возвращается к работе (wake\_up).

#### 4.2.2 Формат структуры «IOCtl\_Fimatic\_rdIO»

```
struct IOCtl_Fimatic_rdIO {
    int    NetAddr;      // Структура данных для чтения информации с модулей IO (AI
    int    ChType;        // -> Адрес целевого модуля в крейте модулей IO
    int    ChNumber;      // -> тип запрашиваемых каналов IO в целевом модуле ввода-
    int    retCod;        // -> количество каналов в целевом модуле IO
    char   Data[16];      // -> код возврата. Устанавливается драйвером по итогам вы
    // -> массив из 16 байт. Возвращается от драйвера в UserSp
    char   ChStatus[8];   // модули AI - [4байта(float)/канал]x4 = 16 байт; моду
    int    resTime;       // -> Статус канала в модуле: 0x00-"Отключен", "Ошибка кан
    // -> время в мс (количество системных тиков) ожидания оте
};
```

Поля структуры данных имеют следующие значения:

n	поле	Описание	Прим.
1	NetAddr	Адрес модуля в сети RS485 на шине модулей ввода/ вывода подключенной к процессорному модулю Fimatic МП300. Возможны значения: 1...255	запись
2	ChType	Тип модуля. Возможны значения (ChTYPE_ANG/ ChTYPE_DGT)	запись
3	ChNumber	Количество каналов в модуле ввода/ вывода. Возможны значение: 2, 4, 8	запись
4	retCod	Код возврата драйвера по итогам выполнения команды. В случае успеха «0». Может иметь следующие значения:  (-10) время ожидания ответного пакета от модуля IO истекло, т. е. TimeOut; (-11) в пакете который вернул модуль IO обнаружена ошибка контрольной суммы CRC;  (-1...-8) «ошибка в ответе модуля». Т.е. модуль IO принял запрос, обработал его, но вернул код ошибки. Значение кода соответствует коду принятому для спецификации протокола шины Fimatic, взятому со знаком минус;	чтение
5	Data[16]	Массив из 16 Int значений, в котором драйвер возвращает кодовые значения физических каналов IO. Формат заполнения зависит от типа модуля и количества каналов в нем. Нулевой ячейке массива соответствует канал IO с наименьшим номером в модуле. Положение байт в массиве для Float соответствует «little endian» (IEEE754) Например:  модуль МСАН4. [4 байта float]x[4 канала] = 16 байт; модуль МСД8. [1 байт]x[8 каналов] = 8 байт; модуль МВДО8. [1 байт]x[8 каналов] = 8 байт;  Для части модулей IO старшие элементы массива не используются.	чтение
6	ChStatus[8]	Массив возвращает статусную информацию по каналам целевого модуля IO. Соответствует кодам спецификации протокола Fimatic. Содержит коды типов каналов модулей или их коды статуса («Ошибка канала», «Отключен» и т. д.) Номер элемента массива соответствует номеру канала в модуле.	чтение
7	resTime	Счетчик, показывающий остаток временного интервала до истечения ошибки по Time-Out. В последнем случае данное поле возвращает 0 tic. Исчисляется в тиках системного таймера.	чтение

### 4.2.3 Примеры работы команды

Для получения данных от модуля IO, данный IOCTL системный вызов использует команду протокола Fimatic Bus с кодом 0x41.

На рисунке ниже приводится дамп данных (снятый при помощи USB адаптера с шины Fimatic), полученный при запросе драйвером информации от модуля «МСАТС»:

```
klabukov@klabukov-VB:~/Рабочий стол$ sudo cat /dev/ttyUSB1|hexdump -C
00000000  04 41 00 01 02 03 2d 31  04 41 00 6f fe 7b 86 45  |.A....-1.A.o.{.E|
00000010  01 6f 9b d3 8c 45 02 6f  ff ff 90 45 03 6f 01 00  |.o...E.o...E.o..|
00000020  91 c5 9c b9 04 41 00 01  02 03 2d 31 04 41 00 6f  |.....A....-1.A.o|
```

Пакет запроса, который отправляет драйвер, имеет следующий формат и размер в восемь байт (на рисунке выделен розовым цветом):

- «04»: номер узла сети RS485 на задней шине Fimatic. Определяется в поле «NetAddr» структуры «IOctl\_Fimatic\_rdIO»;
- «41»: код команды согласно спецификации Fimatic Bus;
- «00», «01», «02», «03»: перечисление номеров каналов. Для данного типа модуля их количество равно четырем и было определено в поле «ChNumber», структуры «IOctl\_Fimatic\_rdIO»;
- «2d», «31»: контрольная CRC сумма, которую драйвер автоматически вычисляет и подставляет в конец пакета запроса;

Пакет ответа, который отправляет модуль «МСАТС», имеет следующий формат и размер в 28 байт (на рисунке выделен зеленым цветом):

- «04»: номер узла сети RS485 на задней шине Fimatic, в котором расположен запрашиваемый модуль;
- «41»: код команды согласно спецификации Fimatic Bus;
- «00»: канал аналогового ввода нулевой;
- «6f» код типа модуля согласно спецификации протокола Fimatic Bus;
- «fe», «7b», «86», «45»: измеренное значение в формате Float (четыре байта) с физического канала модуля;
- далее аналогично данные для каналов «01», «02», «03». Все они имеют тип «6f» и четырех байтовое значение в формате Float, измеренное на соответствующем канале модуля;
- «9c», «b9»: контрольная CRC сумма которую модуль «МСАТС» автоматически вычисляет и подставляет в конец ответного пакета (проверяется драйвером для проверки целостности данных);

Следующий дамп демонстрирует информацию, передаваемую по шине Fimatic при запросе драйвером состояния выходных реле модуля «МВДР8»:

```

klabukov@klabukov-VB:~/Рабочий стол$ sudo cat /dev/ttyUSB1 | hexdump -C
00000000  01 41 00 01 02 03 04 05 06 07 1b e0 01 41 00 91 |.A.....A..|
00000010  01 01 91 00 02 91 01 03 91 00 04 91 01 05 91 01 |.....|
00000020  06 91 00 07 91 01 00 69 01 41 00 01 02 03 04 05 |.....i.A.....|
00000030  06 07 1b e0 01 41 00 91 01 01 91 00 02 91 01 03 |....A.....|

```

Расшифровка данных в пакете запроса драйвером:

- «01»: номер узла сети RS485, в поле «NetAddr»;
- «41»: код команды согласно спецификации Fimatic Bus;
- «00», «01», «02», «03», «04», «05», «06», «07»: перечисление запрашиваемых номеров каналов . Для данного типа модуля их количество равно восьми и было определено в поле «ChNumber»;
- «1b», «e0»: контрольная CRC сумма которую драйвер автоматически вычисляет и подставляет в конец пакета запроса;

Пакет ответа, который отправляет модуль «МВДР8», имеет следующий формат и размер в 28 байт (на рисунке выделен зеленым цветом):

- «01»: номер узла сети RS485 на задней шине Fimatic, в котором расположен запрашиваемый модуль;
- «41»: код команды согласно спецификации Fimatic Bus;
- «00»: номер реле в модуле;
- «91» код типа модуля согласно спецификации протокола Fimatic Bus;
- «01»: состояние в которое установлено реле, т. е. «Включено»;
- далее аналогично для всех реле модуля «01» ... «07». Все они имеют тип «91» и значение соответствующего реле «0» или «1». При этом «0» соответствует «Отключено», «1» соответствует «Включено». Таким образом, на момент выполнения запроса реле модуля имели состояния: «1010 1010»;
- «00», «69»: контрольная CRC сумма которую модуль «МВДР8» автоматически вычисляет и подставляет в конец ответного пакета (проверяется драйвером для проверки целостности данных);

## 4.3 Команда «cmdIOctl\_Fimatic\_wrIO» (0x42)

### 4.3.1 Общее описание

Данная команда позволяет управлять дискретными выходными модулями (DO) из линейки модулей Fimatic.

После вызова IOCTL драйвер на основании данных структуры «IOctl\_Fimatic\_wrIO» формирует пакет запроса к модулю, отправляет его, по прерываниям от порта принимает обратно код успешного выполнения команды или код ошибки, помещает статус в структуру «IOctl\_Fimatic\_wrIO» и возвращает пользователю в User Space.



### 4.3.2 Формат структуры «IOCtl\_Fimatic\_wrIO»

```
struct IOCtl_Fimatic_wrIO {           // Структура данных для управления модулем DO (типы
//
    int    NetAddr;                   // -> Адрес целевого Выходного модуля в крейте модуля
    int    ChType;                    // -> тип каналов в целевом модуле Вывода (определяет
    int    ChNumber;                  // -> количество каналов в целевом Выходном модуле
    int    retCod;                    // <- код возврата. Устанавливается драйвером по итс
    char    Data[8];                 // -> массив из 8 байт. Задается в UserSpace и опред
// Нумерация каналов, по индексу массива Data,от
    int    resTime;                  // <- время в мс (количество системных тиков) ожидан
};
```

Поля структуры данных имеют следующие значения:

п	поле	Описание	Прим.
1	NetAddr	Адрес модуля в сети RS485 на шине модулей ввода/ вывода подключенной к процессорному модулю Fimatic МП300. Возможны значения: 1...255	запись
2	ChType	Тип модуля. Для данной команды всегда «2» (ChTYPE_DGT)	запись
3	ChNumber	Количество каналов в модуле ввода/ вывода. Возможны значение: 4, 8	запись
4	retCod	Код возврата драйвера по итогам выполнения команды. В случае успеха «0». Может иметь следующие значения:  (-10) время ожидания ответного пакета от модуля IO истекло, т. е. TimeOut; (-11) в пакете который вернул модуль IO обнаружена ошибка контрольной суммы CRC;  (-1...-8) «ошибка в ответе модуля». Т.е. модуль IO принял запрос, обработал его, но вернул код ошибки. Значение кода соответствует коду, принятому для спецификации протокола шины Fimatic, взятому со знаком минус;	чтение
5	Data[8]	Массив из 8 Int значений, в котором драйвер возвращает значения физических каналов DO. Формат заполнения зависит от типа модуля и количества каналов в нем. Нулевой ячейке массива соответствует канал DO с наименьшим номером в модуле. Например:  модуль МВДР8. [1 байт]x[8 каналов] = 8 байт; модуль МВДО8.[1 байт]x[8 каналов] = 8 байт;  «0» соответствует команде: «Включить». «1» соответствует команде «Отключить».  Для четырехканальных дискретных модулей старшие элементы массива не используются.	чтение
6	resTime	Счетчик, показывающий остаток временного интервала до истечения ошибки по Time-Out. В последнем случае данное поле возвращает 0 tic. Исчисляется в тиках системного таймера.	чтение

### 4.3.3 Пример работы команды

Для управления модулем DO, данный IOCTL системный вызов использует команду протокола Fimatic Bus с кодом 0x42.

На рисунке ниже приводится дамп данных (снятый при помощи USB адаптера с шины Fimatic), полученный при управлении реле в модуле «МВДР8»:

```
klabukov@klabukov-VB:~/Рабочий стол$ sudo cat /dev/ttyUSB1|hexdump -C
[sudo] пароль для klabukov:
00000000  01 42 00 91 01 01 91 00 02 91 01 03 91 00 04 91 |.B.....|
00000010  01 05 91 01 06 91 00 07 91 01 ff 22 01 42 80 11 |.....".B..|
00000020  01 42 00 91 01 01 91 00 02 91 01 03 91 00 04 91 |.B.....|
```

Пакет запроса на управление реле, который отправляет драйвер, имеет следующий формат и размер в 28 байт (на рисунке выделен розовым цветом):

- «01»: номер узла сети RS485 на задней шине Fimatic. Определяется в поле «NetAddr» структуры «IOctl\_Fimatic\_wrIO»;
- «42»: код команды согласно спецификации Fimatic Bus;
- «00»: номер канала (реле) в модуле DO;
- «91»: код типа модуля согласно спецификации протокола Fimatic Bus;
- «01»: значение в которое нужно установить соответствующий канал модуля дискретного вывода, например в данном примере «Включить»;
- далее аналогично данные для каналов «01» ... «07». Все они имеют тип «91» и один байт который содержит: «0» для команды «Отключить» или «1» для команды «Включить» соответствующий канал DO;
- «ff», «22»: контрольная CRC сумма, которую драйвер автоматически вычисляет и подставляет в конец пакета запроса;

Пакет ответа (квитации успешного выполнения команды), который отправляет модуль «МВДР8», имеет следующий формат и размер в четыре байта (на рисунке выделен зеленым цветом):

- «01»: номер узла сети RS485 на задней шине Fimatic, в котором расположен запрашиваемый модуль;
- «42»: код выполненной команды в случае успеха;
- «80», «11»: контрольная CRC;

## 4.4 Команды IOCTL для работы со счетчиками ошибок

### 4.4.1 Общее описание диагностики драйвера

Как было написано выше, для основных системных вызовов IOCTL («IOctl\_Fimatic\_rdIO», «IOctl\_Fimatic\_wrIO») драйвер возвращает в пространство пользователя коды ошибок. Это позволяет приложению пользователя обрабатывать эти ошибки и вести их учет. Для удобства в драйвер также встроены сервисные функции по ведению и управлению собственными счетчиками ошибок. Они работают в пространстве ядра на уровне модуля драйвера, а их значения могут быть считаны программой пользователя.

Счетчики существуют для модулей IO, расположенных на задней шине RS485 по адресам: 1...31. С каждым из перечисленных адресов связаны отдельные счетчики по событиям:

- превышение времени ожидания ответа (TimeOut);
- ошибка контрольной суммы (CRC);
- ошибка в ответе модуля;
- общий счетчик ошибок (равен сумме перечисленных выше счетчиков)

Например при попытке прочитать с четырехканального модуля «MCATC» несуществующие каналы (поле ChNumber в запросе равно «8» вместо «4»), модуль вернет код ошибки 0x03 («Значение, содержащееся в поле данных запроса, является недопустимой величиной»). При этом драйвер вернет пользователю код (-3) и дополнительно увеличит собственный счетчик по данному виду ошибок для узла с номером 04.

```
klabukov@klabukov-VB:~/Рабочий стол$ sudo cat /dev/ttyUSB1|hexdump -C
[sudo] пароль для klabukov:
00000000 04 41 00 01 02 03 04 05 06 07 0b f0 04 c1 03 21 |.A.....!|
00000010 90 04 41 00 01 02 03 04 05 06 07 0b f0 04 c1 03 |..A.....|
```

В дампе выше (ответ модуля в зеленой рамке): «c1» соответствует команде «0x41» со сброшенным старшим битом. Код ошибки которую вернул модуль «03». Два последних байта это CRC сумма.

Для счетчиков соответствующие IOCTL вызовы позволяют:

- выбрать конкретный номер модуля (его адрес 1 ... 31) со счетчиками которого мы собираемся работать ;
- прочитать номер выбранного для работы со счетчиками модуля;
- прочитать для выбранного номера значения всех его четырех счетчиков;
- сбросить все счетчики для выбранного номера;
- сбросить все счетчики для всех модулей в диапазоне адресов 1...31;

Все IOCTL вызовы для работы со счетчиками драйвера описаны в файле «Fimatic\_Drv.h» файле.

Первые четыре IOCTL команды в качестве аргумента принимают номер счетчика в формате Integer:

```
// Выбрать номер слота диагностики (сделать его текущим)
#define cmdIOCtl_Dgn_ChooseSlot      _IOW(IOC_MAGIC, 20, int)

// Прочитать номер текущего слота диагностики
#define cmdIOCtl_Dgn_SlotCurrent     _IOR(IOC_MAGIC, 21, int)

// Сбросить счетчики текущего слота диагностики
#define cmdIOCtl_Dgn_ResCnt          _IOW(IOC_MAGIC, 22, int)

// Сбросить счетчики всех слотов диагностики
#define cmdIOCtl_Dgn_ResAll          _IOW(IOC_MAGIC, 23, int)
```

Для чтения счетчиков конкретного модуля необходимо выбрать его командой: «cmdIOCtl\_Dgn\_ChooseSlot» и после этого выполнить команду «cmdIOCtl\_Dgn\_RdCnt» для чтения. Описание команды и структуры данных для возврата в программу пользователя:

```
// Прочитать счетчики текущего слота диагностики
#define cmdIOCtl_Dgn_RdCnt           _IOR(IOC_MAGIC, 24, struct IOCtl_Fimatic_Dgn_RdCnt)

struct IOCtl_Fimatic_Dgn_RdCnt {      // Структура счетчиков диагностики. Для текущего слота диагностики
//
    int    cntErr;                    // <- Err: общий счетчик ошибок
    int    cntErr_to;                 // <- Err: ошибки time-out
    int    cntErr_crc;                // <- Err: ошибки контрольной суммы
    int    cntErr_rc;                 // <- Err: ошибки в ответе модуля IO
};
```

## 5 Утилиты драйвера

В состав пакета программ, поставляемых вместе с драйвером, входит набор сервисных утилит. Данные утилиты позволяют:

- протестировать драйвер перед использованием;
- произвести предварительную диагностику аппаратной части системы;
- на этапе разработки приложений пользователя изучить работу API интерфейса драйвера;

Комплект утилит не предполагает его промышленное использование. Он поставляется как есть, без гарантий и является оценочным набором учебных программ. Утилита «IOFimatic\_Checking.c» позволяет выполнять основные команды при работе с модулями IO: читать данные, используя системный вызов «cmdIOCtl\_Fimatic\_rdIO» и управлять модулями DO, используя вызов «cmdIOCtl\_Fimatic\_wrIO». Утилита «IOFimatic\_DiagnCnt.c» позволяет работать со счетчиками диагностики, встроенными в драйвер.

Перед использованием утилит должен быть: собран аппаратный комплект (модуль МП300, модули ввода-вывода, их шина обмена Fimatic-Bus), а также в среде Linux контроллера МП300 запущен драйвер Fimatic\_Drv.

Утилиты поставляются в исходном коде. Это терминальные программы (процессы) Linux, функционирующие в пространстве пользователя. При их разработке для компиляции

использовался пакет компании Texas Instruments - SDK Linux. Готовые к использованию, исполняемые модули также включены в комплект программ.

5.1 Утилита «IOFimatic\_Checking»

5.1.1 Общее описание утилиты

Программа демонстрирует использование IOCTL вызовов драйвера Fimatic\_Drv: «cmdIOCtl\_Fimatic\_rdIO» и «cmdIOCtl\_Fimatic\_wrIO».

Формат использования утилиты:

IOFimatic\_Checking [ChType] [Cmd] [NetAddr] [ChNumber] [D0] [D1] ... [D7]

Расшифровка опций:

[ChType]            тип модуля (ang - аналоговый; dgt - дискретный)  
[Cmd]             команда. Для аналоговых: чтение - rd. Для дискретных: чтение(запись) - rd(wr)  
  
[NetAddr]          сетевой адрес на Backplane. Число 1...99  
[ChNumber]        количество каналов. Для аналоговых 4(2). Для дискретных 8(4)  
  
[Di]                Только для управления модулями D0(опции: dgt + wr ): Включение - 1 (Отключение - 0)

Примеры вариантов использования:

n	Команда	Описание
1	IOFimatic_Checking ang rd 4 4	прочитать значения и состояние каналов модуля "МСАТС4" (сетевой адрес равен 4)
2	IOFimatic_Checking ang rd 7 4	прочитать значения и состояние каналов модуля "МСАН4" (сетевой адрес равен 7)
3	IOFimatic_Checking dgt rd 2 8	прочитать состояние каналов сигнализации модуля "МСД8" (сетевой адрес равен 2)
4	IOFimatic_Checking dgt wr 1 8 0 0 1 1 0 0 1 1	установить реле каналов управления модуля "МВДР8" в состояние: [off][off][on][on][off][off][on][on] (сетевой адрес равен 1)
5	IOFimatic_Checking dgt rd 1 8	прочитать состояние реле каналов управления модуля "МВДР8" (сетевой адрес равен 1)

### 5.1.2 Пример чтения модуля «МСАН4»

```
root@am335x-evm:~# insmod Fimatic_Drv_Main.ko
[ 191.827824] FimaticDrv: Fimatic Bus Drv. V1.2 2023
[ 191.832840] FimaticDrv: set data exchange rate: 57600
[ 191.838001] FimaticDrv: init Diagnostics...
[ 191.844892] FimaticDrv: the device is compatible with: fimatic_uart ...
[ 191.852241] FimaticDrv: IRQ 35
[ 191.863728] FimaticDrv: function fimatic_uart_init. Platform driver Registered ...
root@am335x-evm:~#
root@am335x-evm:~# ./IOFimatic_Checking ang rd 7 4
program run ...
Format: IOFimatic_Checking [ang] [rd] [addr] [chnum] ...
ChType = 0x01
NetAddr = 7
ChNumber = 4
Data[0] = 0x6f 0xe7 0xfb 0x38
Data[1] = 0xab 0xf1 0x32 0x3a
Data[2] = 0xe7 0xd0 0xa2 0x38
Data[3] = 0x41 0x35 0x1e 0x39
ChStatus[0] = 0x42
ChStatus[1] = 0x42
ChStatus[2] = 0x42
ChStatus[3] = 0x42
retCod = 0
resTime = 0x0a
root@am335x-evm:~#
```

### 5.1.3 Пример чтения модуля «МВДО8»

```
root@am335x-evm:~# ./IOFimatic_Checking dgt rd 3 8
program run ...
Format: IOFimatic_Checking [dgt] [rd] [addr] [chnum] ...
ChType = 0x02
NetAddr = 3
ChNumber = 8
Data[0] = 0
Data[1] = 0
Data[2] = 1
Data[3] = 1
Data[4] = 0
Data[5] = 0
Data[6] = 1
Data[7] = 1
ChStatus[0] = 0x91
ChStatus[1] = 0x91
ChStatus[2] = 0x91
ChStatus[3] = 0x91
ChStatus[4] = 0x91
ChStatus[5] = 0x91
ChStatus[6] = 0x91
ChStatus[7] = 0x91
retCod = 0
resTime = 0x0a
root@am335x-evm:~#
```

### 5.1.4 Пример управления модулем «МВДР8»

```
root@am335x-evm:~# ./IOFimatic_Checking dgt wr 1 8 0 0 1 1 0 0 1 1
program run ...
Format: IOFimatic_Checking [dgt] [rd] [addr] [chnum] [d0] ... [d7]
ChType = 0x02
NetAddr = 1
ChNumber = 8
retCod = 0
resTime = 0x09
root@am335x-evm:~#
```

## 5.2 Утилита «IOFimatic\_DiagnCnt»

Программа демонстрирует использование IOCTL вызовов драйвера Fimatic\_Drv обеспечивающих работу со встроенными диагностическими счетчиками.

Формат использования утилиты и расшифровка опций:

IOFimatic\_DiagnCnt [Cmd] [DgnSlot]

[Cmd] Команды

ChooseSlot - установить номер текущего слота диагностики  
SlotCurrent - прочитать номер текущего слота диагностики  
ResCnt - сбросить счетчики текущего слота диагностики  
ResAll - сбросить счетчики всех слотов диагностики  
RdCnt - прочитать состояние счетчиков диагностики для текущего слота  
RdCntAll - прочитать состояние счетчиков диагностики для всех слотов

[DgnSlot] номер слота для команды 0...31

В примере ниже пользователь после запуска драйвера:

- читает несуществующий модуль по адресу 10. Счетчик «TimeOut» для данного адреса увеличивается на один;

```
root@am335x-evm:~# ./IOFimatic_Checking ang rd 10 4
program run ...
Format: IOFimatic_Checking [ang] [rd] [addr] [chnum] ...
ChType = 0x01
NetAddr = 10
ChNumber = 4
Data[0] = 0xff 0xff 0xff 0xff
Data[1] = 0xff 0xff 0xff 0xff
Data[2] = 0000 0000 0000 0000
Data[3] = 0000 0000 0000 0000
ChStatus[0] = 0000
ChStatus[1] = 0000
ChStatus[2] = 0000
ChStatus[3] = 0000
retCod = -10
resTime = 0000
root@am335x-evm:~#
```



- читает из модуля «МСАТП», расположенного по адресу 5, несуществующие каналы (восемь вместо четырех). Счетчик «Ошибка в ответе модуля» для данного адреса увеличивается на один;

```
root@am335x-evm:~# ./IOFimatic_Checking ang rd 5 8
program run ...
Format: IOFimatic_Checking [ang] [rd] [addr] [chnum] ...
ChType = 0x01
NetAddr = 5
ChNumber = 8
Data[0] = 0xff 0xff 0xff 0xff
Data[1] = 0xff 0xff 0xff 0xff
Data[2] = 0000 0000 0000 0000
Data[3] = 0000 0000 0000 0000
ChStatus[0] = 0000
ChStatus[1] = 0000
ChStatus[2] = 0000
ChStatus[3] = 0000
retCod = -3
resTime = 0000
root@am335x-evm:~#
```

- ВЫВОДИТ значения всех счетчиков;

```
root@am335x-evm:~# ./IOFimatic_DiagnCnt RdCntAll
Read counters in the All diagnostic slot
Dgn Slot: 0 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 1 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 2 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 3 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 4 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 5 - cntErr: 1 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 1
Dgn Slot: 6 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 7 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 8 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 9 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 10 - cntErr: 1 - cntErr_to: 1 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 11 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 12 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
```

- выбирает для работы со счетчиками модуль «МСАТП», расположенный по адресу 5. Проверяет, что он выбран. Читает значение счетчиков для выбранного модуля;

```
root@am335x-evm:~# ./IOFimatic_DiagnCnt ChooseSlot 5
Change Dgn Slot to: 5
root@am335x-evm:~# ./IOFimatic_DiagnCnt SlotCurrent
Current Dgn Slot: 5
root@am335x-evm:~# ./IOFimatic_DiagnCnt RdCnt
Read counters in the current diagnostic slot
Dgn Slot: 5
cntErr: 1
cntErr_to: 0
cntErr_crc: 0
cntErr_rc: 1
root@am335x-evm:~#
```



- сбрасывает счетчики для модуля «МСАТП» по адресу 5. Читает значение счетчиков для выбранного модуля;

```
root@am335x-evm:~# ./IOFimatic_DiagnCnt ResCnt
Clear current Dgn Slot ...
root@am335x-evm:~# ./IOFimatic_DiagnCnt RdCnt
Read counters in the current diagnostic slot
Dgn Slot: 5
cntErr: 0
cntErr_to: 0
cntErr_crc: 0
cntErr_rc: 0
root@am335x-evm:~#
```

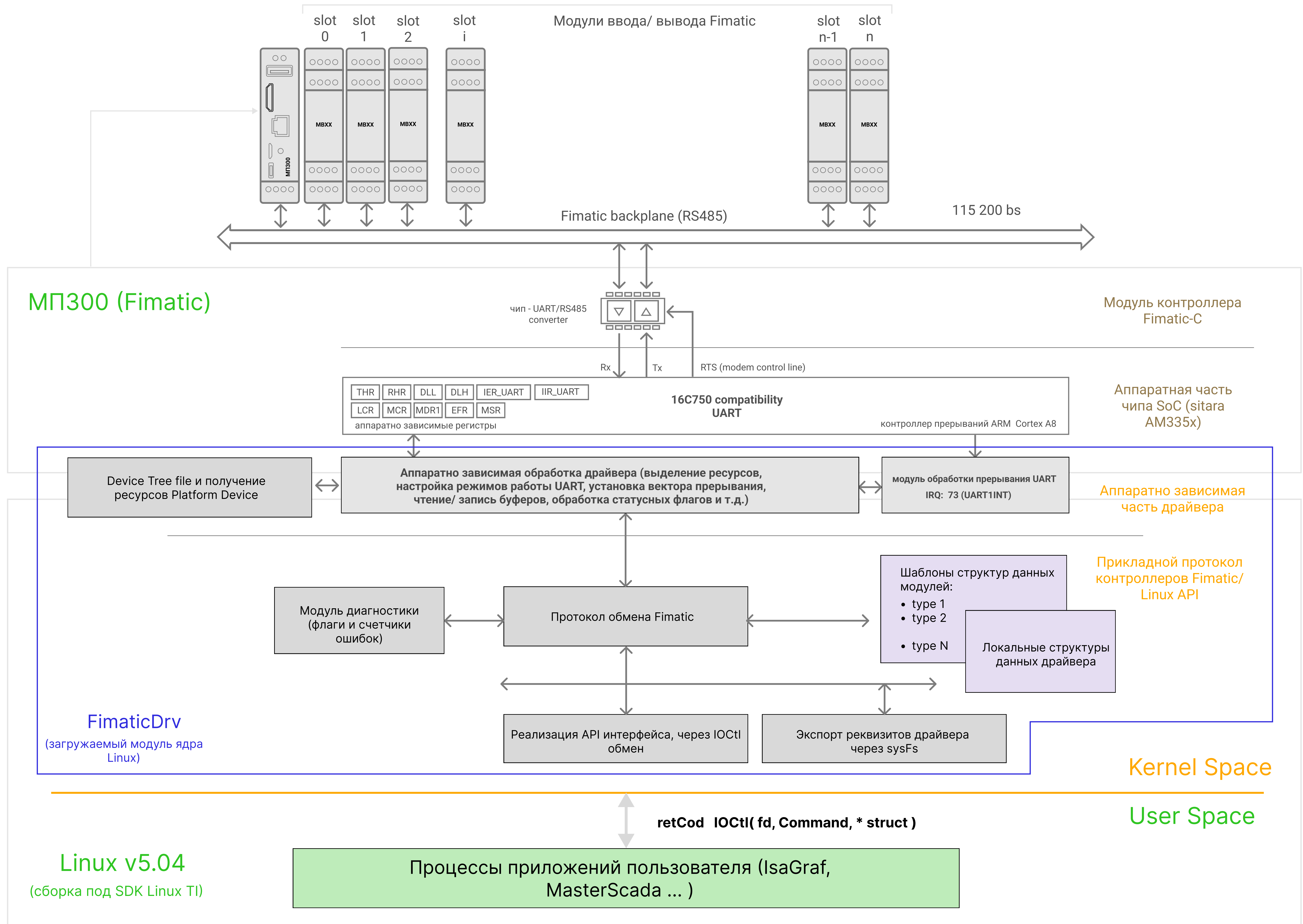
- сбрасывает все счетчики. Проверяет все счетчики;

```
root@am335x-evm:~# ./IOFimatic_DiagnCnt ResAll
Clear All Dgn Slot ...
root@am335x-evm:~# ./IOFimatic_DiagnCnt RdCntAll
Read counters in the All diagnostic slot
Dgn Slot: 0 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 1 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 2 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 3 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 4 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 5 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 6 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 7 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 8 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
Dgn Slot: 9 - cntErr: 0 - cntErr_to: 0 - cntErr_crc: 0 - cntErr_rc: 0
```

Приложение №1. Эскиз структуры драйвера

Приложение №2. Форматы пакетов протокола обмена

Драйвер модулей ввода-вывода линейки контроллеров Fimatic для операционной системы Linux



## Приложение №2. Форматы пакетов протокола обмена

```
// --- Опрос модулей. Команда с кодом 0x41 ---
//
// Модуль: МСД8 (сигнализации дискретный, 8 каналов)
-> 02 41 00 01 02 03 04 05 06 07 eb ef
<- 02 41 00 11 00 01 11 00 02 11 00 03 11 00 04 11 00 05 11 00 06 11 00 07 11 00
cc 1c

// Модуль: МСАН4 (аналоговый напряжение, 4 канала)
-> 07 41 00 01 02 03 2d 02
<- 07 41 00 42 f3 d2 4d 39 01 42 2b 5c 4f 3a 02 42 9f ef 87 39 03 42 47 b6 33 39
88 b1

// Модуль: МСАТ4 (аналоговый токовый, 4 канала)
-> 06 41 00 01 02 03 2c d3
<- 06 41 00 46 3f 0a 57 b9 01 46 5e 8f c2 b8 02 46 0c d7 23 b7 03 46 2a 5c 0f b9
50 9a

// Модуль: МСАТП4 (аналоговый термopара, 4 канала)
-> 05 41 00 01 02 03 2c e0
<- 05 41 00 5f 01 00 00 c3 01 5f 01 00 00 c3 02 5f 01 00 00 c3 03 5f 01 00 00 c3
0f d8

// Модуль: МСАТС4 (аналоговый термосопротивление, 4 канала)
-> 04 41 00 01 02 03 2d 31
<- 04 41 00 6f 5f 1f 8a 45 01 6f dc 01 8a 44 02 6f ff ff 90 45 03 6f 01 00 91 c5
ff f0

// Модуль: МВД08 (дискретный выходной, релейный, 8 каналов)
-> 03 41 00 01 02 03 04 05 06 07 ba 2a
<- 03 41 00 91 00 01 91 00 02 91 00 03 91 00 04 91 00 05 91 00 06 91 00 07 91 00
62 4f

// Модуль: МВДР8 (дискретный выходной, транзистор, 8 каналов)
-> 01 41 00 01 02 03 04 05 06 07 1b e0
<- 01 41 00 91 00 01 91 00 02 91 00 03 91 00 04 91 00 05 91 00 06 91 00 07 91 00
91 f0


// --- Управление модулями. Команда с кодом 0x42 ---
//
// Модуль: МВДР8 (дискретный выходной, транзистор, 8 каналов)
-> 01 42 00 91 00 01 91 00 02 91 00 03 91 00 04 91 00 05 91 00 06 91 00 07 91 01
af 7b
<- 01 42 80 11

// Модуль: МВД08 (дискретный выходной, релейный, 8 каналов)
-> 03 42 00 91 00 01 91 00 02 91 00 03 91 00 04 91 00 05 91 00 06 91 00 07 91 01
5c c4
<- 03 42 81 71
```