

УТВЕРЖДАЮ

Генеральный директор

ООО «НОВОМАР»

\_\_\_\_\_ Т.В. Буга

«\_\_\_\_» \_\_\_\_\_ 2021 г.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «А429 КОНСОЛЬНЫЕ ТЕСТЫ»

Модулей

“PCIe-429UD88”

“mPCIe-429UD84”

“ХМС-429UDxx”

**(OCWINDOWS)**

**(7, 8, 8.1, 10)**

Руководство оператора

ЛИСТ УТВЕРЖДЕНИЯ

RU.MCKЮ.15203-01 34 01 -ЛУ

От

Инженер-программист

\_\_\_\_\_  
«\_\_\_\_» \_\_\_\_\_ 2021 г.

\_\_\_\_\_  
«\_\_\_\_» \_\_\_\_\_ 2021 г.

Инев. № подл	Подп. и
Взам. инв. №	Подп. и
Инев. № дубл	Подп. и

Из	Под	Дат
----	-----	-----

Литера

Утвержден  
RU.MCKЮ.15203-01 34 01-ЛЮ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «А429 КОНСОЛЬНЫЕ ТЕСТЫ»

Модулей  
“PCIe-429UD88”  
“mPCIe-429UD84”  
“ХМС-429UDxx”

**(OCWINDOWS)**  
**(7, 8, 8.1, 10)**

Руководство оператора

RU.MCKЮ.15203-01 34 01

Листов 38

2021

Инев. № подл	Подп. и	Взам. инв. №	Инев. № дубл	Подп. и

Из	Под	Дат

Литера

## АННОТАЦИЯ

В книге описываются технологические принципы, использованные в программном обеспечении «А429 Консольные тесты». В частности, рассмотрены функциональное назначение и область применения, условия выполнения, сообщения оператору.

<i>Из</i>	<i>Под</i>	<i>Дат</i>

## СОДЕРЖАНИЕ

СПИСОК СОКРАЩЕНИЙ.....	5
1 НАЗНАЧЕНИЕ ПРОГРАММЫ .....	6
2 УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ .....	7
3 ВЫПОЛНЕНИЕ ПРОГРАММЫ.....	8
3.1 Общее описание работы ПО.....	8
3.2 Запуск и завершение работы ПО. ....	9
3.2.1 Порядок включения. ....	9
3.2.2 Порядок выключения.....	9
3.3 Контроль ошибок линии. ....	9
3.4 Параметры конфигурации тестов. ....	10
3.4.1 Конфигурация теста ctest1.....	10
3.4.2 Конфигурация теста ctest2.....	11
3.4.3 Конфигурация тестов «ctest31»... «ctest36».....	11
4 Описание тестовых программ .....	12
4.1 ctest31 Тест проверки правильности приема и передачи данных для режимов работы передатчика 0 и 3 .....	12
4.2 ctest32 Тест проверки правильности установки бит четности приема и передачи данных для режимов работы передатчика 0 и 3.....	13
4.3 ctest33 Тест проверки правильности фильтрации поля адреса приемника для режимов работы передатчика 0 и 3. ....	15
4.4 ctest34 Тест проверки правильности фильтрации поля идентификатора приемника для режимов работы передатчика 0 и 3.....	16
4.5 ctest351 Тест проверки правильности работы передатчика с однократным запуском в режиме 1. ....	17
4.6 ctest352 Тест проверки правильности работы передатчика в непрерывном цикле в режиме 1. ....	18
4.7 ctest353 Тест проверки правильности работы передатчика в режиме 1 в непрерывном цикле с использованием признака NEW_BIT.....	20
4.8 ctest354 Тест проверки правильности обработки передатчиком в режиме 1 полей дескрипторов РТО и РТР.....	21
4.9 ctest355 Тест проверки правильности работы таймера RRT передатчика в режиме 1. ....	22

Из	Под	Дат

4.10	ctest356	Тест проверки правильности работы передатчика в режиме 1 с использованием разовой команды «запрос» .....	23
4.11	ctest357	Тест проверки правильности работы передатчика в режиме 1 с использованием разовой команды «готовность».....	24
4.12	ctest36	Тест проверки правильности работы передатчика в режиме 2....	26
4.13	ctest1	Тест проверки правильности приема и передачи данных для режимов работы передатчика 0 и 3. ....	31
4.14	ctest2	Тест проверки правильности приема и передачи данных для режима работы передатчика 1. ....	31
4.15	arinc429_dlltest	Тест проверки динамической библиотеки. ....	32
ПРИЛОЖЕНИЕ А Примеры работы тестов.....			33
ПРИЛОЖЕНИЕ Б Коды ошибок. ....			35
ПРИЛОЖЕНИЕ В Схемы тестовых заглушек. ....			36

<i>Из</i>	<i>Под</i>	<i>Дат</i>

## СПИСОК СОКРАЩЕНИЙ

DMA – ПДП (прямой доступ к памяти);

ПО – программное обеспечение;

ПЭВМ – персональная электронная вычислительная машина;

ПК – последовательный код по ГОСТ 18977-79 и РТМ1495-75(последовательный интерфейс ARINC-429);

РК – Разовые Команды по ГОСТ 18977-79(дискретный канал).

<i>Из</i>	<i>Под</i>	<i>Дат</i>

## 1 НАЗНАЧЕНИЕ ПРОГРАММЫ

Программное обеспечение «А429 Консольные тесты» (далее – ПО) служит как пример использования библиотеки взаимодействия и драйвера. Обеспечивает простой «вход в технологию» разработки с использованием библиотеки взаимодействия и драйвера.

ПО позволяет проводить тестирование модулей: «mPCIe-429UD84», «PCIe-429UD88» и «ХМС-429UDxx» (далее «xxxx-429UDxx»).

ПО обеспечивает выполнение следующих основных задач:

- тестирование в режиме передачи fifo;
- тестирование в режиме передачи с помощью микропрограммы;
- тестирование в режиме внутреннего loopback;
- тестирование внутреннего коммутатора каналов ARINC-429;
- тестирование каналов РК и прерываний;
- ведение лога тестирования.

<i>Из</i>	<i>Под</i>	<i>Дат</i>

## 2 УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

2.1 ПО написано на языке программирования «С++» и предназначено для работы в операционной системе Windows.

2.2 Основные требования к конфигурации ПЭВМ и настройкам операционной системы, выполнение которых необходимо для нормальной работы комплекса программ ПО, приведены в таблице.

Таблица 1 – Требования к конфигурации ПЭВМ

Параметр	Необходимый минимум	Рекомендуемый минимум
Тип процессора	Core 2 Duo	Core 2 Duo
Объем оперативной памяти, Мбайт	512	2048
Разрешение экрана	800×600	1024×768
Размер системного шрифта, точек на дюйм	96	120
Цветовая палитра экрана	High-color (16 бит)	High-color (16 бит)

2.3 На модуле должны быть установлены тестовые заглушки:

Таблица 2 – Соответствие тестов, заглушек и модулей.

Модуль	«ctest1», «ctest2», «ctest31»...«ctest357»	«ctest36»
заглушка-«PCIe-429UD88»	МСКЮ.685669.001	МСКЮ.685669.002
заглушка-«mPCIe-429UD84»	МСКЮ.685667.005	МСКЮ.685667.004

Схемы заглушек приведены в Приложении В.

Для работы с модулями «ХМС-429UDxx» используйте адаптер «ХМС-429UDxx-CROSS» с заглушками, соответствующими «PCIe-429UDxx».

Из	Под	Дат



### 3 ВЫПОЛНЕНИЕ ПРОГРАММЫ

#### 3.1 Общее описание работы ПО.

ПО предназначено для тестирования модулей «xxx-429UDxx» и состоит из 14 тестовых программ.

Конфигурация тестов «ctest1» и «ctest2» хранится в каталоге «\config». Конфигурация тестов «ctest31»...«ctest36» задаётся в параметрах командной строки.

Файлы микропрограмм контроллера ARINC-429 находятся в подкаталоге «\mem».

Каждая тестовая программа сохраняет журнал тестирования. Файлы журнала являются обыкновенными текстовыми файлами и записываются в подкаталог «\logs». Файлы журнала имеют автоматически определённое название, которое создано по формуле:

«год-месяц-день\_час\_минута\_секунда\_названиетеста\_идентификатор.log»

<i>Из</i>	<i>Под</i>	<i>Дат</i>

### 3.2 Запуск и завершение работы ПО.

#### 3.2.1 Порядок включения.

Для запуска программы необходимо открыть терминал и написать команду «ctest1 <config>», или «ctest2<config>», или, например, «ctest351» с параметрами.

#### 3.2.2 Порядок выключения.

Для выключения ПО нажмите комбинацию клавиш «CTRL + C».

Тесты «ctest1»и «ctest2» работают бесконечно до принудительного выключения.

Тесты «ctest31»...«ctest36» останавливаются сами по завершению или принудительно.

### 3.3 Контроль ошибок линии.

Контроль ошибок линии осуществляется посредством контроля ряда признаков транзакции слова 2 в блоке DMA для приёмника ARINC-429: биты 21, 22, 23 (см. Приложение Б настоящего документа и Руководство по программированию п.6.3.1.2).

<i>Из</i>	<i>Под</i>	<i>Дат</i>

### 3.4 Параметры конфигурации тестов.

Конфигурации тестов хранятся в текстовых файлах в подкаталоге «config».

В комплекте ПО приложены файлы «ctest1.txt» и «ctest2.txt». Пользователь может изменять существующие и создавать новые файлы конфигурации тестов.

#### 3.4.1 Конфигурация теста ctest1.

Конфигурация теста в файле «ctest1.txt».

```
s 1 //скорость: 0 - 100; 1 - 12-14; 2 - 50
p 1 //чётность: 0; 1
r 0 //реверс: 0; 1
m 0 //режим: 0 - mode 0; 3 - mode 3
t 3 //таймаут: N seconds
c 50 //количество сообщений, отправленных за такт: N штук
```

*Рисунок 3.4.1 – Конфигурация для ctest1.*

Параметры теста, устанавливаются для всех последовательных каналов ARINC-429 в системе:

**s** – скорость приёма-передачи (0 – 100 кбит/с, 1 – 12-14 кбит/с, 2 – 50 кбит/с);

**p** – бит чётности (0 – дополнение до 0, 1 - дополнение до 1);

**r** – реверсбит в первом байте сообщения (0 – прямая передача «1..8,9,10..31», 1 – реверсивная передача «8..1,9,10..31»);

**m** – режим работы передатчика (0 – режим 0, 3 – режим 3);

**t** – таймаут между тактами обмена и выдачи информации на экран, в секундах;

**c** – количество сообщений, выданных за такт (1 - 255);

<i>Из</i>	<i>Под</i>	<i>Дат</i>

### 3.4.2 Конфигурация теста ctest2.

Конфигурация теста в файле «ctest2.txt».

```
s 1 //скорость: 0 - 100; 1 - 12-14; 2 - 50
p 1 //чётность: 0; 1
r 0 //реверс: 0; 1
t 6 //таймаут: N seconds
```

*Рисунок 3.4.2 – Конфигурация для ctest2.*

Параметры теста:

**s** – скорость приёма-передачи (0 – 100 кбит/с, 1 – 12-14 кбит/с, 2 – 50 кбит/с);

**p** – бит чётности (0 – дополнение до 0, 1 - дополнение до 1);

**r** – реверсбит в первом байте сообщения (0 – прямая передача «1..8,9,10..31», 1 – реверсивная передача «8..1,9,10..31»);

**t** – таймаут между тактами обмена и выдачи информации на экран, в секундах;

**c** – количество сообщений, выданных за такт (1 - 255);

### 3.4.3 Конфигурация тестов «ctest31»... «ctest36».

Конфигурация тестов задаётся в параметрах командной строки.

«ctest31 SpeedReceive SpeedSendTransmitModeIsStopByError»

**SpeedReceive**–скоростьприёмников: 0 – 100 кбит/с, 1 – 12-14 кбит/с, 2 – 50 кбит/с, 3 – 12.5 кбит/с, 4 – 14 кбит/с (параметры 3 и 4 доступны только для «ctest31»)

**SpeedSend**–скорость передатчиков: 0 – 100 кбит/с, 1 – 12-14 кбит/с, 2 – 50 кбит/с, 3 – 12.5 кбит/с, 4 – 14 кбит/с (параметры 3 и 4 доступны только для «ctest31»)

**TransmitMode**–режим работы передатчика (только для тестов «ctest31», «ctest32», «ctest33», «ctest34»): 0 – mode 0; 3 – mode 3 (параметр доступен только для «ctest31» ... «ctest34»)

**IsStopByError**–поведение теста при ошибке: 0 –продолжать; 1–останавливать

*Рисунок3.4.3 – Параметры конфигурации для«ctest31»... «ctest36».*

Из	Под	Дат

## 4 Описание тестовых программ

### 4.1 ctest31 Тест проверки правильности приема и передачи данных для режимов работы передатчика 0 и 3

Тест предназначен для проверки правильности приема и передачи данных, а так же в правильности вывода бит 7 – 0 (0 - 7) принятых и переданных данных.

Пример запуска: «ctest31 43 0 0»

Тест предназначен для проверки правильности приема и передачи данных, а так же в правильности вывода бит 7 – 0 (0 - 7) принятых и переданных данных.

Соединение передатчиков и приемников:

- Для модулей с количеством передатчиков равным количеству приемников соединены одноименные передатчики и приемники (1 – 1, 2 – 2, ... 8 - 8).
- Для модулей с количеством передатчиков меньше количества приемников младший номер передатчика соединен с двумя младшими номерами приемников (1 – 1.2, 2 – 3.4, ... 4 – 7,8).
- Для всех модулей одноименные входы/выходы разовых команд соединены между собой.

Конфигурация регистра RX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	W	-	0	0	W	W	W	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	0	1	0

Конфигурация регистра TX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	W	W	W	0	0	0	0	1	0	1	W	W	W	W	W	W	W	W	W	W	W	W	-	-	-	-	-	-	-

Тест проводится 2 раза для режима 0 и для режима 3. При запуске тест конфигурирует приемники и передатчики. Биты RX\_EN и TX\_EN установлены в 1. Данные передаются путем записи в регистр TX\_FIFO\_REG.

Условия теста:

При передаче данных биты 31 - 8, имеют произвольное (случайное) значение, биты 7 – 0 последовательно принимают все допустимые значения (инкремент).

Результат тестирования:

Проверка правильности (сравнение принятых/переданных данных) принятых бит 31 – 8 принятого слова данных, а так же правильность и порядок (зеркальность) принятых бит 7 – 0 (0 – 7) в зависимости от установки битов TX\_REVERCE и RX\_REVERCE.

<i>Из</i>	<i>Под</i>	<i>Дат</i>

#### 4.2 ctest32 Тест проверки правильности установки бит четности приема и передачи данных для режимов работы передатчика 0 и 3.

Программа предназначена для тестирования правильности установки бит четности приёма и передачи данных в режимах работы передатчика 0 и 3

Пример запуска: «ctest3211 0 0»

Тест предназначен для проверки правильности приема и передачи данных, а так же в правильности вычислений бита четности.

Соединение передатчиков и приемников:

- Для модулей с количеством передатчиков равным количеству приемников соединены одноименные передатчики и приемники (1 – 1, 2 – 2, ... 8 - 8).
- Для модулей с количеством передатчиков меньше количества приемников младший номер передатчика соединен с двумя младшими номерами приемников (1 – 1.2, 2 – 3.4, ... 4 – 7,8).
- Для всех модулей одноименные входы/выходы разовых команд соединены между собой.

Конфигурация регистра RX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	W	W	1	-	0	0	W	W	W	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	0	1	0

Конфигурация регистра TX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	W	W	1	W	W	0	0	0	0	1	0	1	W	W	W	W	W	W	W	W	W	W	W	W	-	-	-	-	-	-	-

Тест проводится 2 раза для режима 0 и для режима 3. При запуске тест конфигурирует приемники и передатчики. Биты RX\_EN и TX\_EN установлены в 1. Данные передаются путем записи в регистр TX\_FIFO\_REG.

Условия теста:

При передаче данных биты 31 - 0, имеют произвольное (случайное) значение. Последовательно перебираются значения TX\_PARCHECK, TX\_PARITY, RCV\_PARCHECK, RCV\_PARITY для приемника и передатчика с целью проверки всех возможных сочетаний.

Для каждой проверки слово данных передается два раза: биты 30 – 0 совпадают для обоих слов, бит 31 передается в первом слове как 0, во втором как 1.

При включенном бите TX\_PARCHECK передатчик игнорирует бит 31 записанного слова и вместо него подставляет вычисленный бит четности, полученное значение выдается в DMA.

При выключенном бите TX\_PARCHECK передатчик выдает записанные в него данные вне зависимости от значения бита TX\_PARITY.

В свою очередь приемник при выключенном бите RCV\_PARCHECK принимает данные из линии в том виде, в котором он их получил.

<i>Из</i>	<i>Под</i>	<i>Дат</i>

При включенном бите RCV\_PARCHECK, приемник записывает в бит 31 принятого слова значение 0, если вычисленная четность совпадает с принятым битом 31 (независимо от его значения), или 1, если не совпадает.

Во втором служебном слове DMA приемника бит RX\_PAR\_ERR устанавливается только при включенном бите RCV\_PARCHECK и несовпадении четности. Биты 30 – 0 данных приемника и передатчика должны совпадать во всех случаях.

Результат тестирования:

Таблица соответствия передаваемых и принимаемых данных.

TX_PARC HECK	TX_PAR ITY	RCV_PARC HECK	RCV_PARI TY	№ слова	Бит31 TXWR	Бит31 TX DMA	Бит31 RXDMA
0	0	0	0	1	0	0	0
0	0	0	0	2	1	1	1
0	1	0	0	1	0	0	0
0	1	0	0	2	1	1	1
1	0	0	0	1	0	0/1	0/1
1	0	0	0	2	1	0/1	0/1
1	1	0	0	1	0	1/0	1/0
1	1	0	0	2	1	1/0	1/0
0	0	0	1	1	0	0	0
0	0	0	1	2	1	1	1
0	1	0	1	1	0	0	0
0	1	0	1	2	1	1	1
1	0	0	1	1	0	0/1	0/1
1	0	0	1	2	1	0/1	0/1
1	1	0	1	1	0	1/0	1/0
1	1	0	1	2	1	1/0	1/0
0	0	1	0	1	0	0	0/1
0	0	1	0	2	1	1	0/1
0	1	1	0	1	0	0	0/1
0	1	1	0	2	1	1	0/1
1	0	1	0	1	0	0/1	0
1	0	1	0	2	1	0/1	0
1	1	1	0	1	0	1/0	1
1	1	1	0	2	1	1/0	1
0	0	1	1	1	0	0	0/1
0	0	1	1	2	1	1	0/1
0	1	1	1	1	0	0	0/1
0	1	1	1	2	1	1	0/1
1	0	1	1	1	0	0/1	1
1	0	1	1	2	1	0/1	1
1	1	1	1	1	0	1/0	0
1	1	1	1	2	1	1/0	0

Значения в таблице 0/1 и 1/0 означают то, что для переданных слов 1 и 2 значения должны совпадать, т.е. или 0 для слова 1 и 2, или 1 для слова 1 и 2. Различные значения для слов 1 и 2 являются ошибкой.

Из	Под	Дат

### 4.3 ctest33 Тест проверки правильности фильтрации поля адреса приемника для режимов работы передатчика 0 и 3.

Тест предназначен для проверки правильности фильтрации принятых данных по полю адреса в приемнике.

Пример запуска: «ctest3311 0 0»

Соединение передатчиков и приемников:

- Для модулей с количеством передатчиков равным количеству приемников соединены одноименные передатчики и приемники (1 – 1, 2 – 2, ... 8 - 8).
- Для модулей с количеством передатчиков меньше количества приемников младший номер передатчика соединен с двумя младшими номерами приемников (1 – 1.2, 2 – 3.4, ... 4 – 7,8).
- Для всех модулей одноименные входы/выходы разовых команд соединены между собой.

Конфигурация регистра RX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	-	0	0	W	W	W	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	0	0	0

Конфигурация регистра TX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	W	W	0	0	0	0	1	0	1	W	W	W	W	W	W	W	W	W	W	W	W	-	-	-	-	-	-	-

Тест проводится 2 раза для режима 0 и для режима 3. При запуске тест конфигурирует приемники и передатчики. Биты RX\_EN и TX\_EN установлены в 1. Данные передаются путем записи в регистр TX\_FIFO\_REG.

Условия теста:

При передаче данных биты 31 - 8, имеют произвольное (случайное) значение, биты 7 – 0 последовательно принимают все допустимые значения (инкремент).

Для каждого цикла передаваемых данных регистры LBL\_CONF\_REG\_PCI\_x содержит только одну единицу, с последовательным перебором всех 255 значений.

Значение регистра переписывается перед началом следующего цикла передачи.

После записи регистра выдерживается пауза не менее 10 мкс. При этом условия работу приемников и передатчиков (**биты RX\_EN и TX\_EN**) можно не останавливать.

После передачи данных последнего цикла в регистре RX\_CONF\_REG тест устанавливает в 1 бит RCV\_FLTR\_DIS, после чего однократно выполняет цикл передачи.

Результат тестирования:

Проверка правильности приема (сравнение принятых/переданных данных) по разрешенному полю адреса для всех циклов кроме последнего. Определение лишних или отсутствующих принятых данных. В последнем цикле должны быть приняты данные по всем адресам, независимо от значения регистра LBL\_CONF\_REG\_PCI\_x.

Из	Под	Дат



#### 4.4 ctest34 Тест проверки правильности фильтрации поля идентификатора приемника для режимов работы передатчика 0 и 3.

Тест предназначен для проверки правильности фильтрации принятых данных по полю идентификатора в приемнике.

Пример запуска: «ctest3422 0 0»

Соединение передатчиков и приемников:

- Для модулей с количеством передатчиков равным количеству приемников соединены одноименные передатчики и приемники (1 – 1, 2 – 2, ... 8 - 8).
- Для модулей с количеством передатчиков меньше количества приемников младший номер передатчика соединен с двумя младшими номерами приемников (1 – 1.2, 2 – 3.4, ... 4 – 7,8).
- Для всех модулей одноименные входы/выходы разовых команд соединены между собой.

Конфигурация регистра RX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	-	W	W	W	W	W	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	1	0	0

Конфигурация регистра TX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	W	W	0	0	0	0	1	0	1	W	W	W	W	W	W	W	W	W	W	W	W	-	-	-	-	-	-	-

Тест проводится 2 раза для режима 0 и для режима 3. При запуске тест конфигурирует приемники и передатчики. Биты RX\_EN и TX\_EN установлены в 1. Данные передаются путем записи в регистр TX\_FIFO\_REG.

Условия теста:

В регистре LBL\_CONF\_REG\_PCI включается фильтрация адресов.

При передаче данных биты 31 - 10, имеют произвольное (случайное) значение, биты 9 – 0 последовательно принимают все допустимые значения (инкремент).

После каждого цикла передачи приемник останавливается и устанавливается следующее значение бит RCV\_SDI. Всего выполняется 4 цикла передачи.

После 4 цикла бит RCV\_DECODE регистра RX\_CONF\_REG устанавливается в 0 без остановки приемника и выполняется еще один цикл передачи данных.

Результат тестирования:

Проверка правильности приема (сравнение принятых/переданных данных) по разрешенным полям адреса и идентификатора для всех циклов кроме последнего. Определение лишних или отсутствующих принятых данных. В последнем цикле должны быть приняты данные по разрешенным полям адреса с любым идентификатором.

Из	Под	Дат

#### 4.5 ctest351 Тест проверки правильности работы передатчика с однократным запуском в режиме 1.

Тест предназначен для проверки правильности работы передатчика с однократным запуском в режиме 1 (работа с внутренней памятью передатчика).

Пример запуска: «ctest35111 0»

Соединение передатчиков и приемников:

- Для модулей с количеством передатчиков равным количеству приемников соединены одноименные передатчики и приемники (1 – 1, 2 – 2, ... 8 - 8).
- Для модулей с количеством передатчиков меньше количества приемников младший номер передатчика соединен с двумя младшими номерами приемников (1 – 1.2, 2 – 3.4, ... 4 – 7,8).
- Для всех модулей одноименные входы/выходы разовых команд соединены между собой.

Конфигурация регистра RX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	-	0	0	W	W	W	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	0	1	0

Конфигурация регистра TX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	0	1	0	0	0	0	1	0	1	W	W	W	W	W	W	W	W	W	W	W	W	-	-	-	-	-	-	-

После конфигурации передатчика, перед запуском теста в память дескрипторов всех передатчиков (TX\_DESC\_RAM) загружается содержимое файла desk\_test\_01\_0.mem. Память данных передатчиков (TX\_DATA\_RAM) заполняется произвольным образом для удобства контроля (сравнения) данных принятых приемниками.

##### Условия теста этап 1:

Для запуска теста после разрешения работы передатчика в регистр TX\_CONTROL\_REG записывается бит TX\_START\_ONES установленный в 1.

Тест последовательно передает 256 слов данных из области памяти данных передатчика. После выдачи последнего слова данных передатчик останавливает работу и сбрасывает бит TX\_START\_ONES в состояние 0.

##### Условия теста этап 2:

После окончания теста этапа 1, без остановки работы передатчика, в память дескрипторов передатчика по адресу равному половине адресного пространства памяти данного канала (для адресов 8400...87FCh – адрес 8600) записывается значение 0. В регистр TX\_CONTROL\_REG записывается бит TX\_START\_ONES установленный в 1. Тест последовательно передает 128 слов данных из области памяти данных передатчика. После выдачи последнего слова данных передатчик останавливает работу и сбрасывает бит TX\_START\_ONES в состояние 0.

##### Результат тестирования:

Проверка правильности приема 256 слов данных после этапа 1. Проверка правильности приема 128 слов данных после этапа 2. Проверка сброса бита TX\_START\_ONES.

Из	Под	Дат

#### 4.6 ctest352 Тест проверки правильности работы передатчика в непрерывном цикле в режиме 1.

Тест предназначен для проверки правильности работы передатчика в непрерывном цикле режима 1 (работа с внутренней памятью передатчика).

Пример запуска: «ctest352 1 1 0»

Соединение передатчиков и приемников:

- Для модулей с количеством передатчиков равным количеству приемников соединены одноименные передатчики и приемники (1 – 1, 2 – 2, ... 8 - 8).
- Для модулей с количеством передатчиков меньше количества приемников младший номер передатчика соединен с двумя младшими номерами приемников (1 – 1.2, 2 – 3.4, ... 4 – 7,8).
- Для всех модулей одноименные входы/выходы разовых команд соединены между собой.

Конфигурация регистра RX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	-	0	0	W	W	W	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	0	1	0

Конфигурация регистра TX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	0	1	0	0	0	0	1	0	1	W	W	W	W	W	W	W	W	W	W	W	W	-	-	-	-	-	-	-

После конфигурации передатчика, перед запуском теста в память дескрипторов всех передатчиков (TX\_DESC\_RAM) загружается содержимое файла desk\_test\_01\_0.mem. Память данных передатчиков (TX\_DATA\_RAM) заполняется произвольным образом для удобства контроля (сравнения) данных принятых приемниками.

##### Условия теста этап 1:

Для запуска теста после разрешения работы передатчика в регистр TX\_CONTROL\_REG записывается бит TX\_START\_STOP установленный в 1.

Тест в цикле передает по 256 слов данных из области памяти данных передатчика. Достаточно выполнить порядка 10 циклов, или предусмотреть в тесте ключи для задания количества циклов.

##### Условия теста этап 2:

После окончания теста этапа 1, в регистр TX\_CONTROL\_REG записывается бит TX\_START\_STOP установленный в 0, в память дескрипторов передатчика по адресу равному половине адресного пространства памяти данного канала (для адресов 8400...87FCh – адрес 8600) записывается значение 00000000h. В регистр TX\_CONTROL\_REG записывается бит TX\_START\_STOP установленный в 1. Тест в цикле передает по 128 слов данных из области памяти данных передатчика.

##### Условия теста этап 3:

После окончания теста этапа 2, в регистр TX\_CONTROL\_REG записывается бит TX\_START\_STOP установленный в 0, в память дескрипторов передатчика по адресу равному половине адресного пространства памяти данного канала (для адресов 8400...87FCh – адрес

Из	Под	Дат

8600) записывается значение 00A8010h. В регистр TX\_CONTROL\_REG записывается бит TX\_START\_STOP установленный в 1. Тест в цикле передает по 256 слов данных из области памяти данных передатчика с интервалом в 10 мс между 128 и 130 словами данных.

Результат тестирования:

Проверка правильности приема N циклов по 256 слов данных после этапа 1. Проверка правильности приема N циклов по 128 слов данных после этапа 2. Проверка наличия временного интервала между 128 и 130 словом данных на этапе 3.

<i>Из</i>	<i>Под</i>	<i>Дат</i>

#### 4.7 ctest353 Тест проверки правильности работы передатчика в режиме 1 в непрерывном цикле с использованием признака NEW\_BIT.

Тест предназначен для проверки правильности работы передатчика в непрерывном цикле режима 1 (работа с внутренней памятью передатчика) и использованием признака NEW\_BIT.

Пример запуска: «ctest352 1 1 0»

Соединение передатчиков и приемников:

- Для модулей с количеством передатчиков равным количеству приемников соединены одноименные передатчики и приемники (1 – 1, 2 – 2, ... 8 - 8).
- Для модулей с количеством передатчиков меньше количества приемников младший номер передатчика соединен с двумя младшими номерами приемников (1 – 1.2, 2 – 3.4, ... 4 – 7,8).
- Для всех модулей одноименные входы/выходы разовых команд соединены между собой.

Конфигурация регистра RX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	-	0	0	W	W	W	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	0	1	0

Конфигурация регистра TX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	0	1	0	0	0	0	1	0	1	W	W	W	W	W	W	W	W	W	W	W	W	-	-	-	-	-	-	-

После конфигурации передатчика, перед запуском теста в память дескрипторов всех передатчиков (TX\_DESC\_RAM) загружается содержимое файла desk\_test\_02\_0.mem. Память данных передатчиков (TX\_DATA\_RAM) заполняется произвольным образом для удобства контроля (сравнения) данных принятых приемниками.

##### Условия теста этап 1:

Для запуска теста после разрешения работы передатчика в регистр TX\_CONTROL\_REG записывается бит TX\_START\_STOP установленный в 1.

Тест передает 256 слов данных из области памяти данных передатчика, после чего передача данных прекращается.

##### Условия теста этап 2:

После окончания теста этапа 1, в регистр TX\_DATA\_RAM по произвольным адресам записывается одно или несколько слов данных. Передатчик передает вновь записанные данные. После передачи данных процедура повторяется. Тест проводится с последовательной записью данных по всем адресам TX\_DATA\_RAM.

##### Результат тестирования:

Проверка правильности приема 256 слов данных после этапа 1. Проверка правильности передачи только вновь записанных данных после этапа 2.

Из	Под	Дат

#### 4.8 ctest354 Тест проверки правильности обработки передатчиком в режиме 1 полей дескрипторов РТО и РТР.

Тест предназначен для проверки правильности обработки передатчиком полей дескрипторов РТО и РТР в непрерывном цикле режима 1 (работа с внутренней памятью передатчика).

Пример запуска: «ctest354 1 1 0»

Соединение передатчиков и приемников:

- Для модулей с количеством передатчиков равным количеству приемников соединены одноименные передатчики и приемники (1 – 1, 2 – 2, ... 8 - 8).
- Для модулей с количеством передатчиков меньше количества приемников младший номер передатчика соединен с двумя младшими номерами приемников (1 – 1.2, 2 – 3.4, ... 4 – 7,8).
- Для всех модулей одноименные входы/выходы разовых команд соединены между собой.

Конфигурация регистра RX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	-	0	0	W	W	W	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	0	1	0

Конфигурация регистра TX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	0	1	0	0	0	0	1	0	1	W	W	W	W	W	W	W	W	W	W	W	W	-	-	-	-	-	-	-

После конфигурации передатчика, перед запуском теста в память дескрипторов всех передатчиков (TX\_DESC\_RAM) загружается содержимое файла desk\_test\_03\_0.mem. Память данных передатчиков (TX\_DATA\_RAM) заполняется произвольным образом для удобства контроля (сравнения) данных принятых приемниками.

Условия теста:

Для запуска теста после разрешения работы передатчика в регистр TX\_CONTROL\_REG записывается бит TX\_START\_STOP установленный в 1.

В четных циклах тест передает 128 слов данных из области памяти данных передатчика по четным адресам, в нечетных циклах тест передает 128 слов данных из области памяти данных передатчика по нечетным адресам. Достаточно выполнить 20 циклов.

Результат тестирования:

Проверка правильности приема 128 слов данных по четным адресам и 128 слов данных по нечетным адресам.

<i>Из</i>	<i>Под</i>	<i>Дат</i>

#### 4.9 ctest355 Тест проверки правильности работы таймера RRT передатчика в режиме 1.

Тест предназначен для проверки правильности работы таймера RRT передатчика в непрерывном цикле режима 1 (работа с внутренней памятью передатчика).

Пример запуска: «ctest355 1 1 0»

Соединение передатчиков и приемников:

- Для модулей с количеством передатчиков равным количеству приемников соединены одноименные передатчики и приемники (1 – 1, 2 – 2, ... 8 - 8).
- Для модулей с количеством передатчиков меньше количества приемников младший номер передатчика соединен с двумя младшими номерами приемников (1 – 1.2, 2 – 3.4, ... 4 – 7,8).
- Для всех модулей одноименные входы/выходы разовых команд соединены между собой.

Конфигурация регистра RX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	-	0	0	W	W	W	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	0	1	0

Конфигурация регистра TX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	0	1	0	0	0	0	1	0	1	W	W	W	W	W	W	W	W	W	W	W	W	-	-	-	-	-	-	-

После конфигурации передатчика, перед запуском теста в память дескрипторов всех передатчиков (TX\_DESC\_RAM) загружается содержимое файла desk\_test\_04\_0.mem. Память данных передатчиков (TX\_DATA\_RAM) заполняется произвольным образом для удобства контроля (сравнения) данных принятых приемниками. В регистре TX\_CONTROL\_REG устанавливаются биты TX\_RR и TX\_RRT в зависимости от скорости передачи данных. Значения должны превышать в несколько раз время передачи 16 слов данных для текущей скорости передачи.

Условия теста этап 1:

Для запуска теста после разрешения работы передатчика в регистр TX\_CONTROL\_REG записывается бит TX\_START\_STOP установленный в 1.

Тест в цикле передает по 16 слов данных с длительностью цикла, определяемой регистром TX\_CONTROL\_REG.

Условия теста этап 2:

Без остановки работы передатчика устанавливается в значение 1 бит TX\_SKIP\_RRT регистра TX\_CONTROL\_REG. Тест в цикле передает по 16 слов данных с длительностью цикла, определяемой скоростью передачи данных и паузой между сообщениями.

Результат тестирования:

Проверка правильности приема всех слов данных для этапов 1 и 2. Проверка длительности циклов, получаемая из второго и третьего слова DMA для этапов 1 и 2. Значения TX\_RRT и FREE\_TIMER (из DMA) выводятся в лог программы.

Из	Под	Дат

#### 4.10 ctest356 Тест проверки правильности работы передатчика в режиме 1 с использованием разовой команды «запрос»

Тест предназначен для проверки правильности работы передатчика с использованием разовых команд в режиме 1 (работа с внутренней памятью передатчика).

Тест проводится со стандартным значениям скорости передачи/приема 12 – 14,5 кбит/с. Тест может выполняться последовательно для каждого канала в отдельности.

Пример запуска: «ctest356 1 1 0»

Соединение передатчиков и приемников:

- Для модулей с количеством передатчиков равным количеству приемников соединены одноименные передатчики и приемники (1 – 1, 2 – 2, ... 8 - 8).
- Для модулей с количеством передатчиков меньше количества приемников младший номер передатчика соединен с двумя младшими номерами приемников (1 – 1.2, 2 – 3.4, ... 4 – 7,8).
- Для всех модулей одноименные входы/выходы разовых команд соединены между собой.

Конфигурация регистра RX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	-	0	0	0	1	0	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	0	1	0

Конфигурация регистра TX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-

После конфигурации передатчика, перед запуском теста в память дескрипторов всех передатчиков (TX\_DESC\_RAM) загружается содержимое файла desk\_test\_01\_0.mem. Память данных передатчиков (TX\_DATA\_RAM) заполняется произвольным образом для удобства контроля (сравнения) данных принятых приемниками. В регистре ARINC429\_SC\_INT\_MASK устанавливается значение бит SCx\_FALL = 1.

Условия теста:

Для запуска теста в регистре ARINC429\_SC\_OUT не менее чем на 50 мкс устанавливается значение бит SCx равным 1, затем сбрасывается в 0. При получении прерывания от входных Разовых Команд не менее чем через 100 мкс запускается работа передатчиков записью в регистр TX\_CONTROL\_REG бита TX\_START\_ONES установленного в 1. Тест последовательно передает 256 слов данных из области памяти данных передатчика. После выдачи последнего слова данных передатчик останавливает работу и сбрасывает бит TX\_START\_ONES в состояние 0. Тест повторяется несколько раз для каждого канала передатчика.

Результат тестирования:

Проверка правильности выдачи и приема разовых команд. Проверка прерывания. Проверка правильности логической привязки входов разовых команд к каналу передатчика. Проверка правильности приема 256 слов. Проверка сброса бита TX\_START\_ONES.

Из	Под	Дат



#### 4.11 ctest357 Тест проверки правильности работы передатчика в режиме 1 с использованием разовой команды «ГОТОВНОСТЬ»

Тест предназначен для проверки правильности работы передатчика с использованием разовых команд в режиме 1 (работа с внутренней памятью передатчика).

Тест проводится со стандартным значениям скорости передачи/приема 12 – 14,5 кбит/с. Тест может выполняться последовательно для каждого канала в отдельности.

Пример запуска: «ctest352 1 1 0»

Соединение передатчиков и приемников:

- Для модулей с количеством передатчиков равным количеству приемников соединены одноименные передатчики и приемники (1 – 1, 2 – 2, ... 8 - 8).
- Для модулей с количеством передатчиков меньше количества приемников младший номер передатчика соединен с двумя младшими номерами приемников (1 – 1.2, 2 – 3.4, ... 4 – 7,8).
- Для всех модулей одноименные входы/выходы разовых команд соединены между собой.

Конфигурация регистра RX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	-	0	0	0	1	0	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	0	1	0

Конфигурация регистра TX\_CONF\_REG:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-

После конфигурации передатчика, перед запуском теста в память дескрипторов всех передатчиков (TX\_DESC\_RAM) загружается содержимое файла desk\_test\_01\_0.mem. Память данных передатчиков (TX\_DATA\_RAM) заполняется произвольным образом для удобства контроля (сравнения) данных принятых приемниками. В регистре ARINC429\_SC\_INT\_MASK устанавливается значение бит SCx\_RISE = 1. При этом приемники уже сконфигурированы, но выключены (RX\_EN = 0).

Условия теста:

Для запуска теста в регистре ARINC429\_SC\_OUT программа устанавливает значение бита SCx равным 1. При получении прерывания включает соответствующий канал приемника. Не менее чем через 100 мкс программа запускает работу передатчиков записью в регистр TX\_CONTROL\_REG бита TX\_START\_ONES установленного в 1. Тест последовательно передает 256 слов данных из области памяти данных передатчика. После выдачи последнего слова данных передатчик останавливает работу и сбрасывает бит TX\_START\_ONES в состояние 0. После окончания передачи данных программа устанавливает значение бит SCx равным 0. Тест повторяется несколько раз для каждого канала передатчика.

Результат тестирования:

Проверка правильности выдачи и приема разовых команд. Проверка прерывания. Проверка правильности логической привязки входов и выходов разовых команд к каналу передатчика и

Из	Под	Дат

приемника. Проверка правильности приема 256 слов. Проверка сброса бита TX\_START\_ONES.  
Проверка изменения состояния регистра ARINC429\_SC\_IN.

<i>Из</i>	<i>Под</i>	<i>Дат</i>

## 4.12 ctest36 Тест проверки правильности работы передатчика в режиме 2.

Тест предназначен для проверки правильности работы передатчиков в непрерывном цикле режима 2 (режим работы коммутатора).

Пример запуска: «ctest36 1 1 0»

Соединение передатчиков и приемников:

- Для модулей PCIe-429UD88 передатчик №1 соединен с приемником №1, передатчик №2 соединен с приемником №2, передатчик №3 соединен с приемником №3, передатчик №4 соединен с приемниками №4, №5, №6, №7, №8. Передатчики с номерами №5, №6, №7, №8 соединений с приемниками не имеют.
- Для модулей mPCIe-429UD84 передатчик №1 соединен с приемником №1, передатчик №2 соединен с приемником №2, передатчик №3 соединен с приемником №3, передатчик №4 соединен с приемниками №4, №5, №6, №7, №8.
- Для всех модулей одноименные входы/выходы разовых команд соединены между собой.

Конфигурация регистра RX\_CONF\_REG для всех приемников:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	-	0	0	W	W	W	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	0	0	0

Конфигурация регистра TX\_CONF\_REG для всех передатчиков за исключением №4:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	1	0	0	0	0	0	1	0	1	W	W	W	W	W	W	W	W	W	W	W	W	-	-	-	-	-	-	-

Конфигурация регистра TX\_CONF\_REG для передатчика №4:

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	0	1	0	0	0	0	1	0	1	W	W	W	W	W	W	W	W	W	W	W	W	-	-	-	-	-	-	-

После конфигурации передатчика, перед запуском теста в память данных и дескрипторов всех передатчиков загружается содержимое файлов desk\_test\_xx\_n.mem и dat\_test\_xx\_n.mem.

В память передатчика №1 (№5) загружается содержимое файлов dat\_test\_10\_1.mem и desk\_test\_10\_1.mem. Контроль передаваемых данных осуществляется приемником №1.

В память передатчика №2 (№6) загружается содержимое файлов dat\_test\_10\_2.mem и desk\_test\_10\_2.mem. Контроль передаваемых данных осуществляется приемником №2.

В память передатчика №3 (№7, №8) загружается содержимое файлов dat\_test\_10\_3.mem и desk\_test\_10\_3.mem. Контроль передаваемых данных осуществляется приемником №3.

В память передатчика №4 загружается содержимое файлов dat\_test\_10\_4.mem и desk\_test\_10\_4.mem. Контроль передаваемых данных осуществляется приемником №4.

Примечание 1: контроль передачи данных передатчиками №5 - №8 осуществляется только проверкой DMA передатчиков.

Приемники №5, №6, №7 и №8 используют фильтрацию входных данных и являются источниками данных для передатчиков №1, №2 и №3.

Из	Под	Дат

Перед запуском теста таймер TX\_RRT передатчика №4 программируется на время, превышающее время суммарной передачи 256 слов данных в 1,2 – 1,5 раза, бит TX\_SKIP\_RRT = 0.

Передача всех данных цикла (256 слов без фильтрации или менее, при условии фильтрации) передатчиками №1, №2, №3 осуществляется до начала нового цикла передачи данных передатчиком №4.

Условия теста этап 1:

Передатчик №4 передает данные на все подключенные к нему приемники. Данные организованы таким образом, что байт 2 эквивалентен полю «LABEL» (байт 0), всего передается 256 слов с полем «LABEL» от 00h до FFh и полем SDI = 00b. Принимаемые данные не фильтруются приемниками №5, №6, №7 и №8. Приемники №1, №2, №3 и №4 принимают все поступающие на них данные без фильтрации, полученные данные используются для контроля правильности выполнения теста.

Передатчик №1 передает данные, полученные приемниками №5, №6, №7 и №8 с заменой поля SDI на 01b. Причем диапазон поля «LABEL» 0 – 63 передается из приемника №5, 64 – 127 передается из приемника №6, 128 – 191 передается из приемника №7 и 192 – 255 передается из приемника №8.

Передатчик №2 передает данные, полученные приемниками №5, №6, №7 и №8 с заменой поля SDI на 10b. Причем диапазон поля «LABEL» 0 – 63 передается из приемника №8, 64 – 127 передается из приемника №7, 128 – 191 передается из приемника №6 и 192 – 255 передается из приемника №5.

Передатчик №3 передает данные, полученные приемниками №5, №6, №7 и №8 с заменой поля «LABEL» на инверсное, т.е. значение 01h преобразуется в значение FEh. Причем диапазон поля «LABEL» 0 – 63 передается из приемника №8, 64 – 127 передается из приемника №7, 128 – 191 передается из приемника №6 и 192 – 255 передается из приемника №5.

Условия теста этап 2:

Передатчик №4 передает данные на все подключенные к нему приемники. Данные организованы таким образом, что байт 2 эквивалентен полю «LABEL» (байт 0), всего передается 256 слов с полем «LABEL» от 00h до FFh и полем SDI = 00b. Принимаемые данные фильтруются приемниками №5, №6, №7 и №8 по полю «LABEL». Фильтрация осуществляется для каждого приемника последовательно для диапазонов поля «LABEL» 0 – 63, 64 – 127, 128 – 191, 192 – 255.

Приемники №1, №2, №3 и №4 принимают все поступающие на них данные без фильтрации, полученные данные используются для контроля правильности выполнения теста.

Передатчик №1 передает данные, полученные приемниками №5, №6, №7 и №8 с заменой поля SDI на 01b. Причем диапазон поля «LABEL» 0 – 63 передается из приемника №5, 64 – 127 передается из приемника №6, 128 – 191 передается из приемника №7 и 192 – 255 передается из приемника №8. Данные, отфильтрованные приемниками по полю «LABEL», не передаются.

Передатчик №2 передает данные, полученные приемниками №5, №6, №7 и №8 с заменой поля SDI на 10b. Причем диапазон поля «LABEL» 0 – 63 передается из приемника №8, 64 – 127 передается из приемника №7, 128 – 191 передается из приемника №6 и 192 – 255 передается из приемника №5. Данные, отфильтрованные приемниками по полю «LABEL», не передаются.

Передатчик №3 передает данные, полученные приемниками №5, №6, №7 и №8 с заменой поля «LABEL» на инверсное, т.е. значение 01h преобразуется в значение FEh. Причем диапазон поля «LABEL» 0 – 63 передается из приемника №8, 64 – 127 передается из приемника №7, 128 – 191 передается из приемника №6 и 192 – 255 передается из приемника №5. Данные, отфильтрованные приемниками по полю «LABEL», не передаются.

Из	Под	Дат

Результат тестирования этап 1:

Проверка правильности приема всех данных приемниками №4, №5, №6, №7 и №8 без изменения полей SDI и LABEL. Проверка правильности приема всех данных приемниками №1, №2, №3 с измененными полями SDI и LABEL.

Результат тестирования этап 2:

Проверка правильности приема всех данных приемниками №4, №5, №6, №7 и №8 без изменения полей SDI и LABEL. Проверка правильности приема всех данных приемниками №1, №2, №3 с измененными полями SDI и LABEL, а так же с учетом установленных фильтров для каждого канала приема №5, №6, №7 и №8.

Примечание 2: данные, передаваемые передатчиками №1, №2, №3, №5, №6, №7 и №8, могут передаваться в нескольких циклах не последовательно.

Например:

Цикл 1 – LABEL 0 – 3, LABEL 12 – 32...

Цикл 2 – LABEL 4 – 11, LABEL 33 – 71...

.....

За время суммарного цикла передачи передатчиком №4 передаются все данные, передатчиками №1, №2, №3, №5, №6, №7 и №8 в одном или нескольких циклах для каждого передатчика.

Пояснения к тесту 3.6 в режиме фильтрации данных.

Последовательность фильтрации данных представлена на рис.А.1 и рис.А.2.

<i>Из</i>	<i>Под</i>	<i>Дат</i>

Передатчик 4



Приемник 4



Приемник 5



Приемник 6



Приемник 7



Приемник 8



Передатчик 1,5. Замена SDI из принятого 00 на 01



Передатчик 2,6. Замена SDI из принятого 00 на 10



Передатчик 3,7,8. Замена Label из принятого на инверсное



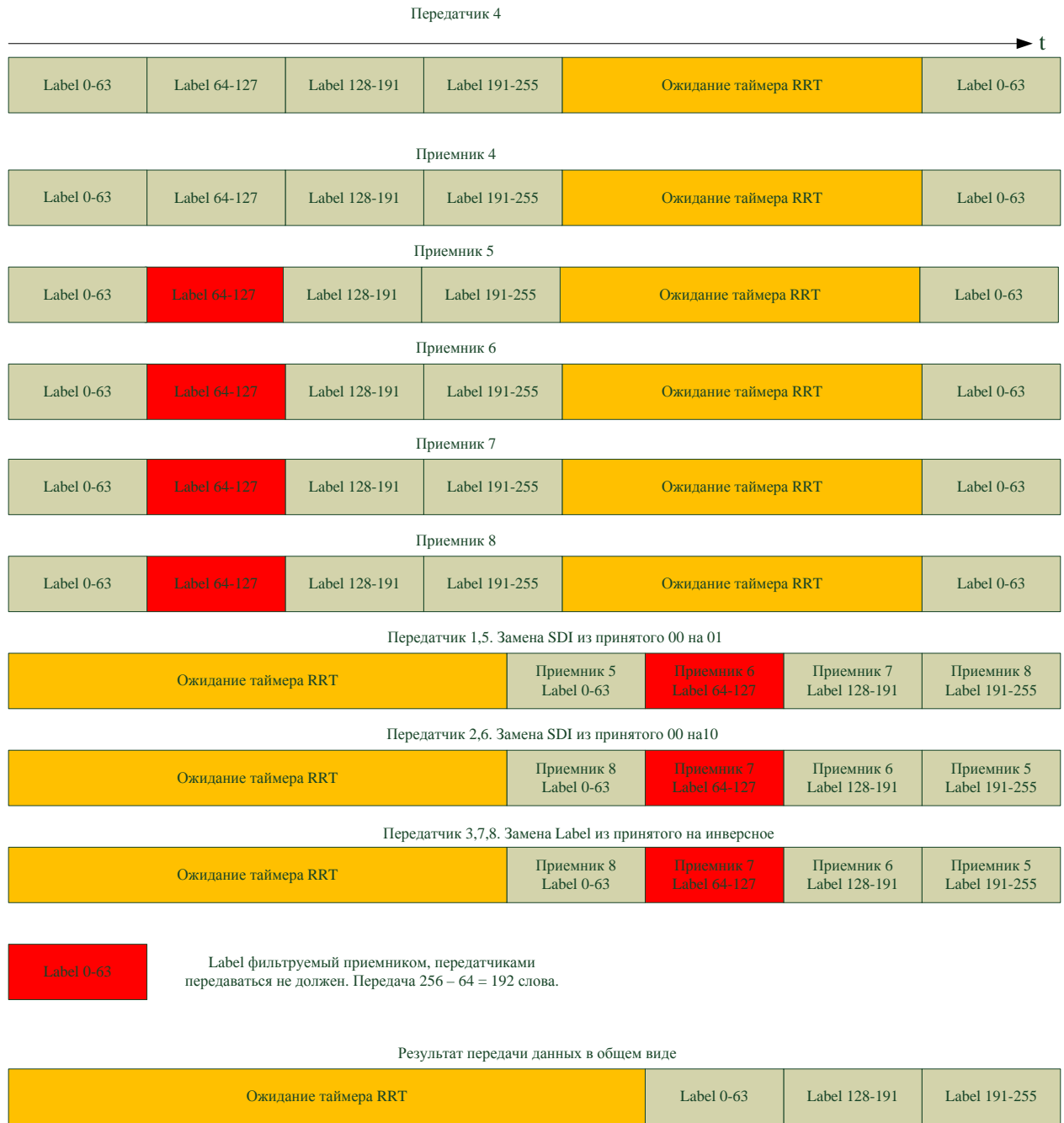
Label фильтруемый приемником, передатчиками передаваться не должен. Передача 256 – 64 = 192 слова.

Результат передачи данных в общем виде



*Рисунок 4.6.1 – Фильтрация данных поля Label 0 – 63.*

<i>Из</i>	<i>Под</i>	<i>Дат</i>



*Рисунок 4.6.2 – Фильтрация данных поля Label 64 – 127.*

Примечание: номера приемников и передатчиков представлены в виде №1 - №8, что соответствует физическим адресам 00h – 07h. Приемники №1 - №4 предназначены для контроля правильности приема/передачи данных, для них фильтры не включаются. Все передатчики, кроме №4 передают только данные с признаком «Newdata» от приемников. **Перед записью новых фильтров приема рекомендуется производить очистку памяти приемников (RCV\_RAM\_CLS). В противном случае (запись нового фильтра во время приема данных) данные могут передаваться некорректно.**

Из	Под	Дат

#### 4.13 ctest1 Тест проверки правильности приема и передачи данных для режимов работы передатчика 0 и 3.

Тест предназначен для проверки правильности приема и передачи данных при работе передатчиков в циклическом режиме FIFO, а также проверке работоспособности линий сухих контактов (разовых команд).

Пример запуска: «ctest1ctest1.txt»

##### Условия теста:

Тест обеспечивает передачу каждым передатчиком блоков данных не менее 32 слов в каждом блоке, с паузой между блоками эквивалентной или превышающей время передачи блока данных. Данные передачи формируются случайным образом. Параллельно с передачей данных производится замыкание/размыкание сухих контактов случайным образом на произвольное время. Временная диаграмма выполнения теста представлена на рис.А.1.

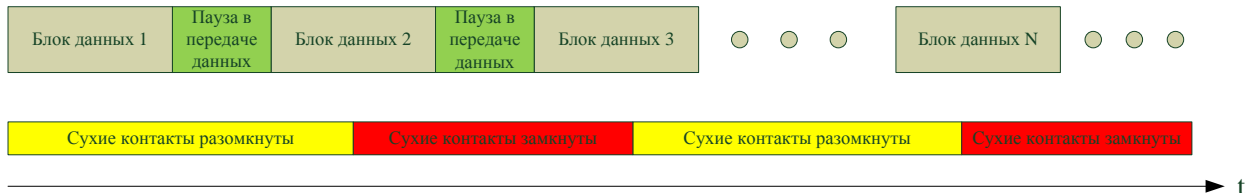


Рисунок А.1. Временная диаграмма теста 1.

#### 4.14 ctest2 Тест проверки правильности приема и передачи данных для режима работы передатчика 1.

Тест предназначен для проверки правильности приема и передачи данных при работе передатчиков в циклическом режиме 1 (режим работы с внутренней памятью), а также для проверки работоспособности линий сухих контактов (разовых команд).

Пример запуска: «ctest2ctest2.txt»

##### Условия теста:

После конфигурации передатчика, перед запуском теста в память дескрипторов всех передатчиков (TX\_DESC\_RAM) загружается содержимое файлов desk\_test\_n.mem. Память данных передатчиков (TX\_DATA\_RAM) заполняется произвольным образом для удобства контроля (сравнения) данных принятых приемниками. Каждый передатчик передает по 32 слова данных в непрерывном цикле. Передаются только вновь записанные данные. Т.е. данные переданные в первом цикле, во втором цикле передаваться не будут, если они не перезаписаны к началу второго цикла. Временная диаграмма выполнения представлена на рис.А.2 Блоки данных передаются с минимальными паузами между циклами в случае непрерывного заполнения памяти данных передатчиков (TX\_DATA\_RAM) новыми данными, или с паузами ожидания нового блока данных между циклами в случае периодической записи.



Рисунок А.2. Временная диаграмма теста 2

Из	Под	Дат



#### 4.15 arinc429\_dlltest Тест проверки динамической библиотеки.

Тест представляет из себя копию теста ctest1, собранного с использованием динамической библиотеки. Сразу после запуска тест выводит в консоль версию и дату сборки библиотеки. В остальном тест полностью дублирует ctest1.

Тест предназначен для проверки правильности приема и передачи данных при работе передатчиков в циклическом режиме FIFO, а также проверке работоспособности линий сухих контактов (разовых команд).

Пример запуска: «arinc429\_dlltest ctest1.txt»

##### Условия теста:

Тест обеспечивает передачу каждым передатчиком блоков данных не менее 32 слов в каждом блоке, с паузой между блоками эквивалентной или превышающей время передачи блока данных. Данные передачи формируются случайным образом. Параллельно с передачей данных производится замыкание/размыкание сухих контактов случайным образом на произвольное время. Временная диаграмма выполнения теста представлена на рис.А.1.

<i>Из</i>	<i>Под</i>	<i>Дат</i>

ПРИЛОЖЕНИЕ А Примеры работы тестов.

На рисунках А.1 и А.2 приведены примеры визуализации ctest1 и ctest2.

```
/dev/a429dev-0-ch-3 - num 3 - T_INFO_CHANNEL_TYPE_RECEIVER
/dev/a429dev-0-ch-2 - num 2 - T_INFO_CHANNEL_TYPE_RECEIVER
/dev/a429dev-0-ch-1 - num 1 - T_INFO_CHANNEL_TYPE_RECEIVER
/dev/a429dev-0-ch-0 - num 0 - T_INFO_CHANNEL_TYPE_RECEIVER
Has founded 28 channels

test is processing...
Channel '/dev/a429dev-1-ch-11' = 0 blocks
Channel '/dev/a429dev-1-ch-10' = 0 blocks
Channel '/dev/a429dev-1-ch-9' = 0 blocks
Channel '/dev/a429dev-1-ch-8' = 0 blocks
Channel '/dev/a429dev-1-ch-7' = 0 blocks
Channel '/dev/a429dev-1-ch-6' = 0 blocks
Channel '/dev/a429dev-1-ch-5' = 0 blocks
Channel '/dev/a429dev-1-ch-4' = 0 blocks
Channel '/dev/a429dev-1-ch-3' = 0 blocks
Channel '/dev/a429dev-1-ch-2' = 0 blocks
Channel '/dev/a429dev-1-ch-1' = 0 blocks
Channel '/dev/a429dev-1-ch-0' = 0 blocks
Channel '/dev/a429dev-0-ch-15' = 0 blocks
Channel '/dev/a429dev-0-ch-14' = 0 blocks
Channel '/dev/a429dev-0-ch-13' = 1 blocks
Channel '/dev/a429dev-0-ch-12' = 0 blocks
Channel '/dev/a429dev-0-ch-11' = 0 blocks
Channel '/dev/a429dev-0-ch-10' = 1 blocks
Channel '/dev/a429dev-0-ch-9' = 0 blocks
Channel '/dev/a429dev-0-ch-8' = 0 blocks
Channel '/dev/a429dev-0-ch-7' = 0 blocks
Channel '/dev/a429dev-0-ch-6' = 0 blocks
Channel '/dev/a429dev-0-ch-5' = 0 blocks
Channel '/dev/a429dev-0-ch-4' = 0 blocks
Channel '/dev/a429dev-0-ch-3' = 0 blocks
Channel '/dev/a429dev-0-ch-2' = 0 blocks
Channel '/dev/a429dev-0-ch-1' = 1 blocks
Channel '/dev/a429dev-0-ch-0' = 0 blocks
Device '/dev/a429dev-1-ch-0' devid= 38008 - checking IN_RK 1= 1, IN_RK 2= 1, IN_RK 3= 0, IN_RK 4= 0
Device '/dev/a429dev-0-ch-0' devid= 38007 - checking IN_RK 1= 1, IN_RK 2= 1, IN_RK 3= 1, IN_RK 4= 1
-has passed 3 seconds-
^2
[1]# Остановлен /ctest1 ctest1.txt
```

Рисунок А.1 – Визуализация ctest1.

Напротив названия каждого канала написано количество сообщений переданных/принятых за один такт, пример строки: «Channel '/dev/a429-0-ch-0' = 32 blocks».

Напротив названия каждой платы написано количество зафиксированных входных разовых команд за один такт, пример строки: «Device '/dev/a429dev-0-ch-0' devid= 38008 – checking IN\_RK 1= 1, IN\_RK 2= 1, IN\_RK 3= 1, IN\_RK 4= 1».

Из	Под	Дат

```

/dev/a429dev-0-ch-5 - num 5 - T_INFO_CHANNEL_TYPE_RECEIVER
/dev/a429dev-0-ch-4 - num 4 - T_INFO_CHANNEL_TYPE_RECEIVER
/dev/a429dev-0-ch-3 - num 3 - T_INFO_CHANNEL_TYPE_RECEIVER
/dev/a429dev-0-ch-2 - num 2 - T_INFO_CHANNEL_TYPE_RECEIVER
/dev/a429dev-0-ch-1 - num 1 - T_INFO_CHANNEL_TYPE_RECEIVER
/dev/a429dev-0-ch-0 - num 0 - T_INFO_CHANNEL_TYPE_RECEIVER
Has founded 28 channels

test is processing...
Channel '/dev/a429dev-1-ch-11' = 32 blocks
Channel '/dev/a429dev-1-ch-10' = 32 blocks
Channel '/dev/a429dev-1-ch-9' = 32 blocks
Channel '/dev/a429dev-1-ch-8' = 32 blocks
Channel '/dev/a429dev-1-ch-7' = 32 blocks
Channel '/dev/a429dev-1-ch-6' = 32 blocks
Channel '/dev/a429dev-1-ch-5' = 32 blocks
Channel '/dev/a429dev-1-ch-4' = 32 blocks
Channel '/dev/a429dev-1-ch-3' = 32 blocks
Channel '/dev/a429dev-1-ch-2' = 32 blocks
Channel '/dev/a429dev-1-ch-1' = 32 blocks
Channel '/dev/a429dev-1-ch-0' = 32 blocks
Channel '/dev/a429dev-0-ch-15' = 32 blocks
Channel '/dev/a429dev-0-ch-14' = 32 blocks
Channel '/dev/a429dev-0-ch-13' = 32 blocks
Channel '/dev/a429dev-0-ch-12' = 32 blocks
Channel '/dev/a429dev-0-ch-11' = 32 blocks
Channel '/dev/a429dev-0-ch-10' = 32 blocks
Channel '/dev/a429dev-0-ch-9' = 32 blocks
Channel '/dev/a429dev-0-ch-8' = 32 blocks
Channel '/dev/a429dev-0-ch-7' = 32 blocks
Channel '/dev/a429dev-0-ch-6' = 32 blocks
Channel '/dev/a429dev-0-ch-5' = 32 blocks
Channel '/dev/a429dev-0-ch-4' = 32 blocks
Channel '/dev/a429dev-0-ch-3' = 32 blocks
Channel '/dev/a429dev-0-ch-2' = 32 blocks
Channel '/dev/a429dev-0-ch-1' = 32 blocks
Channel '/dev/a429dev-0-ch-0' = 32 blocks
-has passed 3 seconds-
^2
[3]+ Остановлен ./ctest2 ctest2.txt

```

*Рисунок а.2 – Визуализация ctest2.*

Напротив названия каждого канала написано количество сообщений переданных/принятых за один такт, пример строки: «Channel '/dev/a429-0-ch-0' = 32 blocks».

<i>Из</i>	<i>Под</i>	<i>Дат</i>

## ПРИЛОЖЕНИЕ Б Коды ошибок.

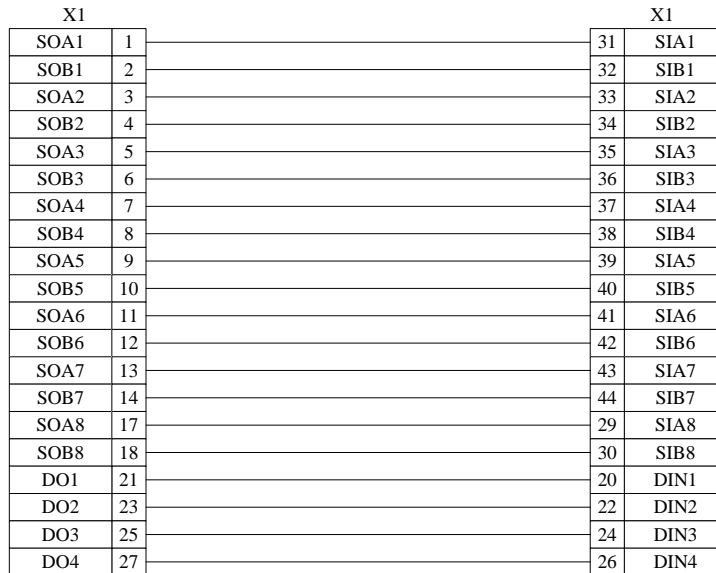
Номер бита	Обозначение	Описание
31	RX	Значение бита отражает источник данных: 0 – приемник.
30	-	Резерв
29-24	RX_BIT_NUM_ERR	Номер бита, принятого с ошибкой кодировки.
23	RX_PAR_ERR	Ошибка четности.
22	RX_GAP_ERR	Ошибка паузы.
21	RX_BIT_ERROR	Ошибка кодировки бита.
20	-	Резерв
19-15	RX_BITLENGTH	Средняя длина принятых бит, измеренная на частоте $10 \cdot F_{пр}$ .
14-0	-	Резерв

Рисунок Б.1 Коды ошибок в слове 2 DMA

Из	Под	Дат

**ПРИЛОЖЕНИЕ В** Схемы тестовых заглушек.

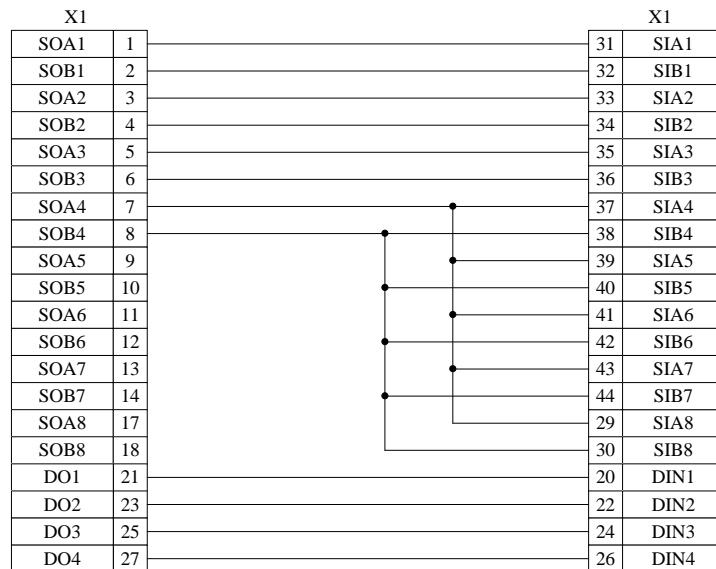
Схема заглушки МСКЮ.685669.001 для модуля «PCIe-429UD88» для работы тестов: «ctest1», «ctest2», «ctest31»... «ctest357».



DB44-M (на кабель)

**Рисунок Б1 – Схема заглушки МСКЮ.685669.001**

Схема заглушки МСКЮ.685669.002 для модуля «PCIe-429UD88» для работы теста «ctest36».



DB44-M (на кабель)

**Рисунок Б2 – Схема заглушки МСКЮ.685667.002**

Из	Под	Дат

Схема заглушки MCKЮ.685667.005 для модуля «mPCIe-429UD84» для работы тестов: «ctest1», «ctest2», «ctest31»... «ctest357» (Может изготавливаться из кабеля MCKЮ.685667.001).

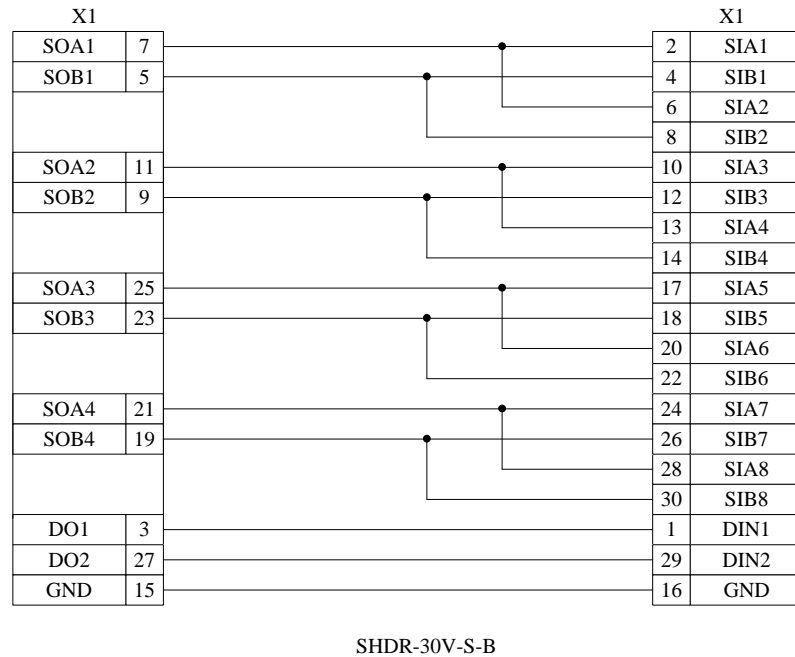


Рисунок Б3 – Схема заглушки MCKЮ.685667.005

Схема заглушки MCKЮ.685667.004 для модуля «mPCIe-429UD84» для работы теста «ctest36» (Может изготавливаться из кабеля MCKЮ.685667.001).

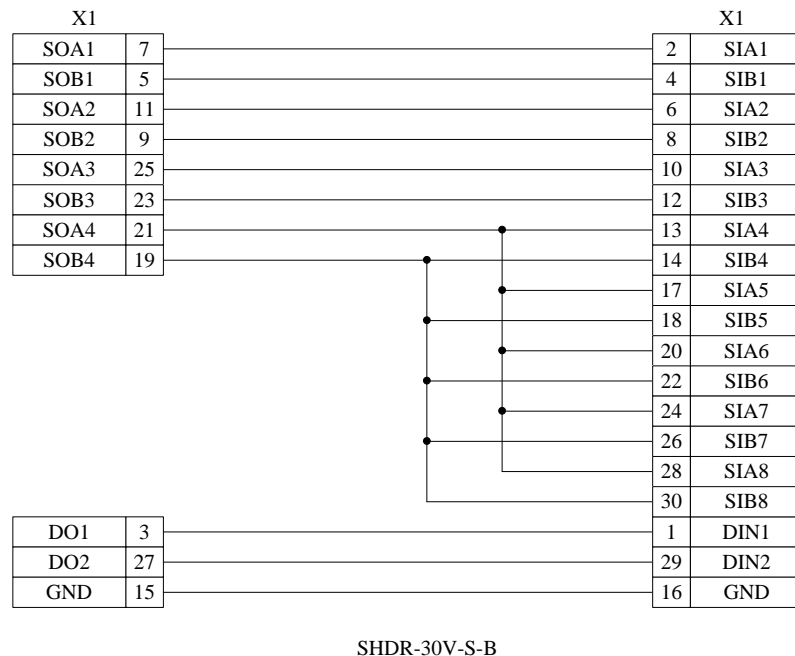


Рисунок Б4 – Схема заглушки MCKЮ. 685667.004

Из	Под	Дат

