

**Инструкция по эксплуатации**  
модуля для сопряжения систем на основе последовательного канала по  
ГОСТ 18977-79 (ARINC 429) с интерфейсом USB 2.0

**ЕСЕ-0206-1**  
ГФКП.468363.024 ИЭ

## Содержание

1. Назначение .....	2
2. Решаемые задачи .....	2
3. Технические характеристики .....	2
4. Структурная схема. ....	3
5. Описание общих принципов работы .....	3
6. Установка модуля.....	4
<b>Описание работы модуля</b>	
7. Описание работы контроллера USB 2.0 .....	4
8. Описание работы контроллера ARINC429 .....	5
9. Работа с выходным каналом .....	6
10. Заполнение буфера BUF256x32. ....	6
11. Настройка режимов выходного канала и старт выдачи .....	8
12. Работа с входными каналами .....	12
13. Формат данных и алгоритм работы CCU в части входных каналов .....	14
13.1 Данные от входных каналов .....	14
13.2 Метки времени.....	15
13.3 Формат данных в конечной точке EP6 IN .....	16
14. Режимы заполнения буфера конечной точки входных каналов EP6 IN ...	18
15. Ошибки входных каналов (ErrorCode).....	19
15. 1 Команда сопровождения ошибки ERRS .....	20
16. Цепи внешнего разъема X1 .....	22

## **1. Назначение**

Модуль **ECE-0206** предназначен для сопряжения интерфейса **USB 2.0** с системами и устройствами по последовательному интерфейсу **ARINC 429 (ГОСТ 18977-79 и РТМ1495-75)**. Модуль выполнен в корпусе размером 198x95x28 мм (герметичный в промышленном исполнении) с внешними интерфейсными разъемами **USB-B** и **DHS-15M (X1)** для подключения до 4-х входных и 1 выходной линий **ARINC 429**.

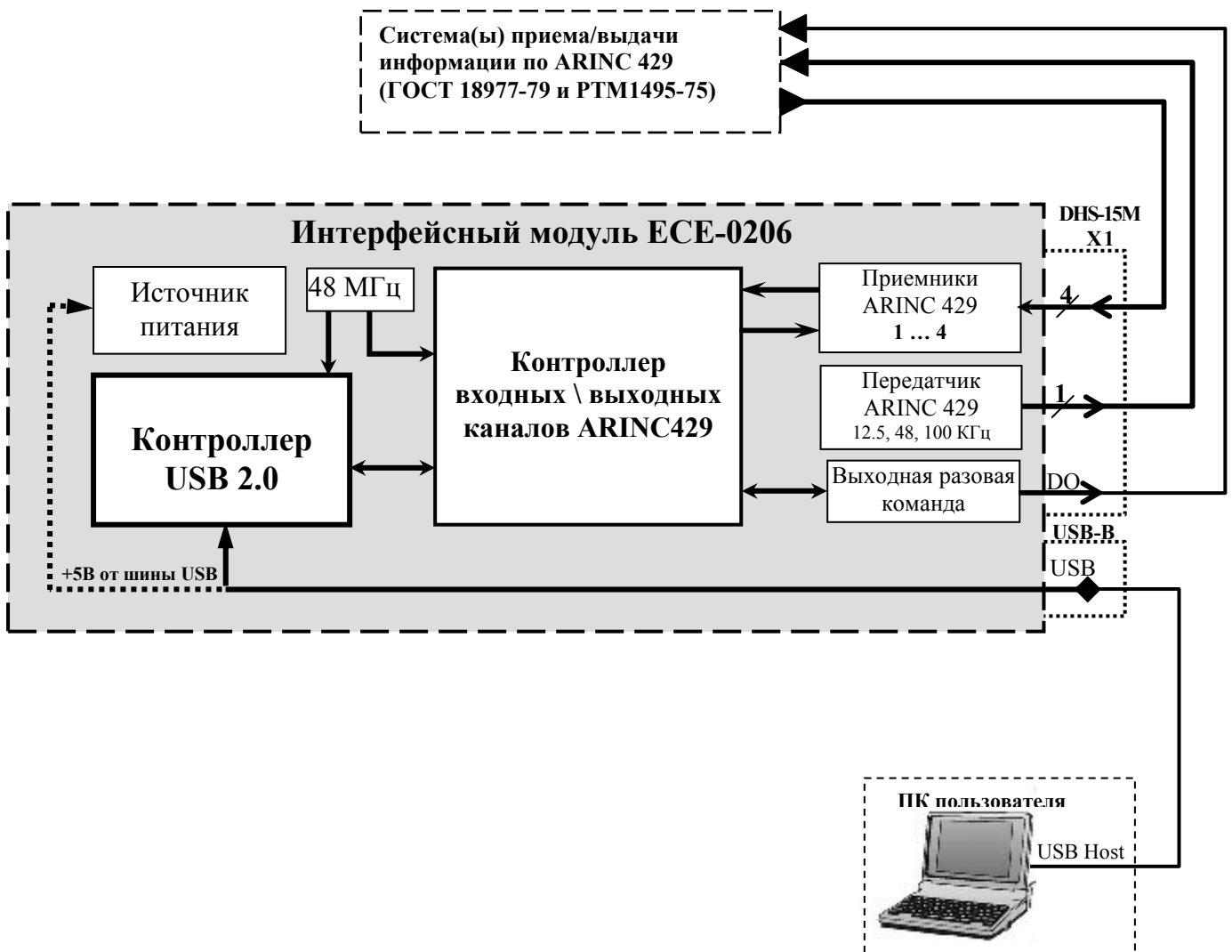
## **2. Решаемые задачи**

- гарантированная доставка цифровой информации приходящей по 4-м независимым каналам ARINC 429 в ПК пользователя.
- сопровождение данных от входных каналов ARINC 429 метками времени.
- формирование кодов ошибок при нарушении протокола ARINC429 в линиях, подключенных к входным каналам.
- загрузка выходной информации с ПК пользователя и формирование выдачи по 1 каналу ARINC 429 в различных режимах.
- выдача разовой команды ТТЛ, а также работа разовой команды в режиме сопровождения выдачи по каналу ARINC 429.

## **3. Технические характеристики**

- Количество входных каналов по ARINC-429: **4;**
- Количество выходных каналов по ARINC-429: **1;**
- Частота приема последовательного кода: **Low – (12..14,5)КГц, Fast – (36..100) КГц;**
- Эквивалентная нагрузка входного канала: **Rн не менее 20 КОм, Сн не более 10 пФ;**
- Частота выдачи последовательного кода: **12,5 КГц +/-1%, 50 КГц +/-1%, 100 КГц +/-1%;**
- Нагрузка выходного канала: **Rн не менее 600 Ом, Сн не более 10.000 пФ;**
- Количество выходных разовых команд типа ТТЛ: **1;**
- Параметры выходной разовой команды: **отсутствие сигнала Uвых<0,4В при In<20мА**  
**наличие сигнала Uвых>2,4В при In<20мА**
- Напряжения питания (от шины USB): **5В +/-5% , I потр. не более 360 мА;**
- Температурные диапазоны от коммерческого **(0 ... +50)** до промышленного **(-20 ... +70);**

#### 4. Структурная схема.



#### 5. Описание общих принципов работы

Устройство ECE-0206 подключается к источникам и/или потребителям информации по стандарту ARINC 429 (ГОСТ 18977-79 и РТМ1495-75) и может являться как имитатором приборов выдающих информацию, так и устройством, обеспечивающим мониторинг и регистрацию данных по 4-м независимым входным каналам.

Все входные слова, приходящие по линиям ARINC 429, поступают на приемники соответствующих каналов (1-4). Приемные каскады преобразуют сигнал от линии ARINC 429 в формат, понятный контроллеру входных/выходных каналов. Затем поступившие данные обрабатываются контроллером и передаются по локальной шине в буферную память контроллера USB. Дальнейшая выборка данных из буфера осуществляется по протоколу USB под управлением Хост контроллера ПК пользователя.

Настройка и заполнение буфера выходного канала ARINC 429 осуществляется аналогично, но в обратной последовательности. Старт выходного канала осуществляется по команде пользователя, при этом загруженная в модуль информация поступает на передатчик ARINC 429 в соответствии с заданным режимом работы. Выходная информация может сопровождаться уровнем выходной разовой команды TTL.

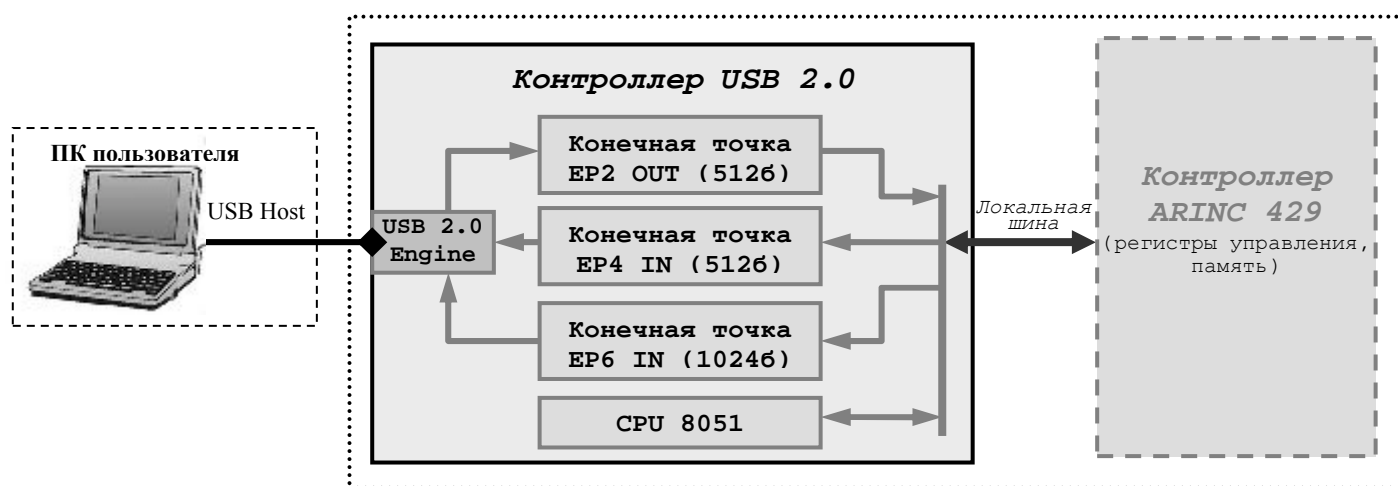
## 6. Установка модуля

Модуль подключается к ПК посредством стандартного кабеля с разъёмами USB-A и USB-B типов. После подключения устройства в операционной системе запускается стандартный механизм инициализации USB устройств. На этом этапе необходимо установить драйверы, поставляемые вместе с модулем. При установке все действия производить в соответствии со стандартными указаниями операционной системы.

## Описание работы модуля

### 7. Описание работы контроллера USB 2.0

Все обмены данными и управление модулем происходят посредством чтения и записи регистров и памяти контроллера каналов ARINC 429. Доступ к контроллеру ARINC429 осуществляется через конечные точки Контроллера USB 2.0. Рассмотрим структурную схему интерфейса контроллера:



EP2, EP4 и EP6 – это конечные точки с двойной буферизацией.

EP2 OUT тип Bulk - предназначена для передачи данных и команд контроллеру.

Через неё происходит запись регистров управления и памяти контроллера ARINC429.

EP4 IN тип Bulk - это конечная точка для чтения значений регистров управления и содержимого памяти контроллера ARINC429. По командам чтения, отправленным через EP2, в EP4 возвращаются запрашиваемые значения.

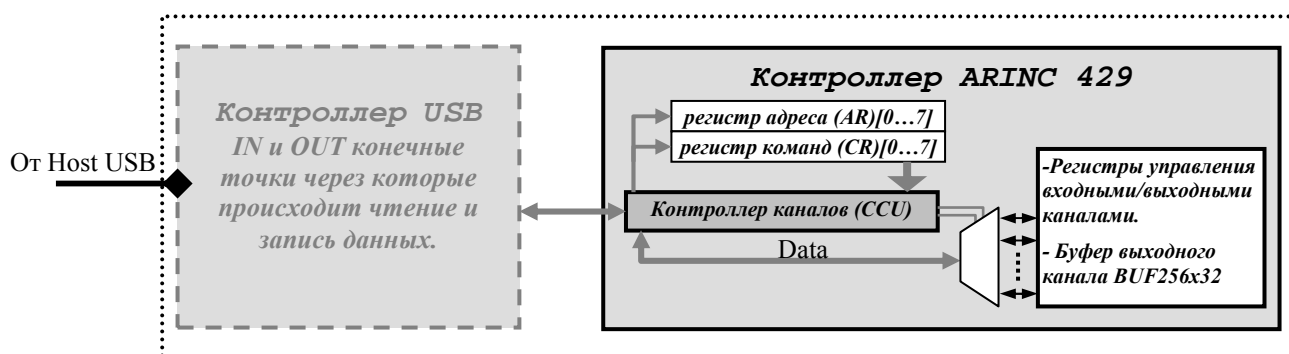
EP6 IN тип Interrupt - предназначена для передачи информации от входных каналов устройства. Все данные, полученные контроллером от 4-х входных каналов ARINC429, сопровождаются метками времени, служебной информацией и записываются в конечную точку EP6. ПО пользователя, работающее с хост контроллером USB, должно непрерывно запрашивать данные у EP6 начиная с момента старта хотя бы одного входного канала.

CPU8051 – это АЛУ с архитектурой 8051 которое выполняет настройку конечных точек при инициализации устройства, а также обработку данных, поступающих от хоста в EP2, и их дальнейшую передачу контроллеру ARINC429 в соответствии с внутренним алгоритмом обмена.

Более подробную информацию о записи и чтении данных через конечные точки можно будет найти в описании соответствующих обменов с контроллером ARINC429.

## 8. Описание работы контроллера ARINC429

Общий принцип обмена данными с контроллером основан на управлении потоком и направлением данных с помощью регистра команд CR (Command Register) и регистром адреса AR (Address Register). На приведенной ниже схеме показано регистровая модель механизма записи/чтения данных.



Любые данные, записываемые в устройство со стороны хоста, должны быть адресованы конечной точке EP2 контроллера USB. После записи в EP2, данные передаются под управлением CPU8051 по локальной шине в контроллер каналов CCU (Channel Control Unit). CCU всегда воспринимает 2 первых байта принятого массива данных как адрес и команду, и записывает их соответственно в регистры AR и CR. Значение в AR указывает адрес буфера памяти выходного канала (0...255). Значение в CR анализируется как команда для CCU. Остальные байты массива воспринимаются как данные для адресуемой области модуля, которая делится на зону регистров управления каналами и буфер выходных каналов BUF256x32 (память на 1Кб с организацией 256x32бит, т.е. адаптированная стандарту ARINC 429)

### Регистр адреса AR:

Adr7	Adr6	Adr5	Adr4	Adr3	Adr2	Adr1	Adr0
AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0

**Adr\*(7..0) – [00h ... FFh].** Адрес буфера выходного канала с организацией 256x32.(см. п. 9)

*\*При значении CR.Com\_Buf = «1» т.е. в режиме доступа к регистрам управления, адрес, записанный в регистр RA значения не имеет.*

### Регистр команд CR:

Blk	Wr_Rd	Com_Buf	X	Ch_adr3	Ch_adr2	Ch_adr1	Ch_adr0
CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

**Blk\*** – блочный режим записи буфера выходного канала.

= «1» - в этом режиме весь массив данных, записанный в конечную точку контроллера USB (до 508 байт\*\*) будет интерпретирован CCU как массив из Nx32бит (где N = от 1 до 127) и записан последовательно с инкрементом счетчика адреса начиная со значения Adr(7..0) в буфер BUF256x32. Поскольку буфер выходного канала BUF256x32 является кольцевым, то при достижении счетчика адреса значения FFh следующая адресуемая ячейка будет иметь адрес 00h, затем 01h и т.д.

= «0» - режим записи 1 слова данных (32 бита).

*\*При Com\_Buf = «1» состояние Blk значения не имеет.*

*\*\* Значение 508 байт обусловлено размером конечной точки 512байт. Т.о. максимальный массив, который можно передать одним пакетом составит: 2 байта - адрес и команда + 127x4=508 байт информации + 2 байта буфера не используются.*

**Wr\_Rd** – выбор операции чтения либо записи адресуемого пространства контроллера

= «1» - чтение данных

= «0» - запись данных

**Com\_Buf** – бит - указатель на адресуемое пространство контроллера

= «1» - запись/чтение данных в/из области регистров управления каналов

= «0» - запись/чтение данных в/из буферной памяти выходного канала

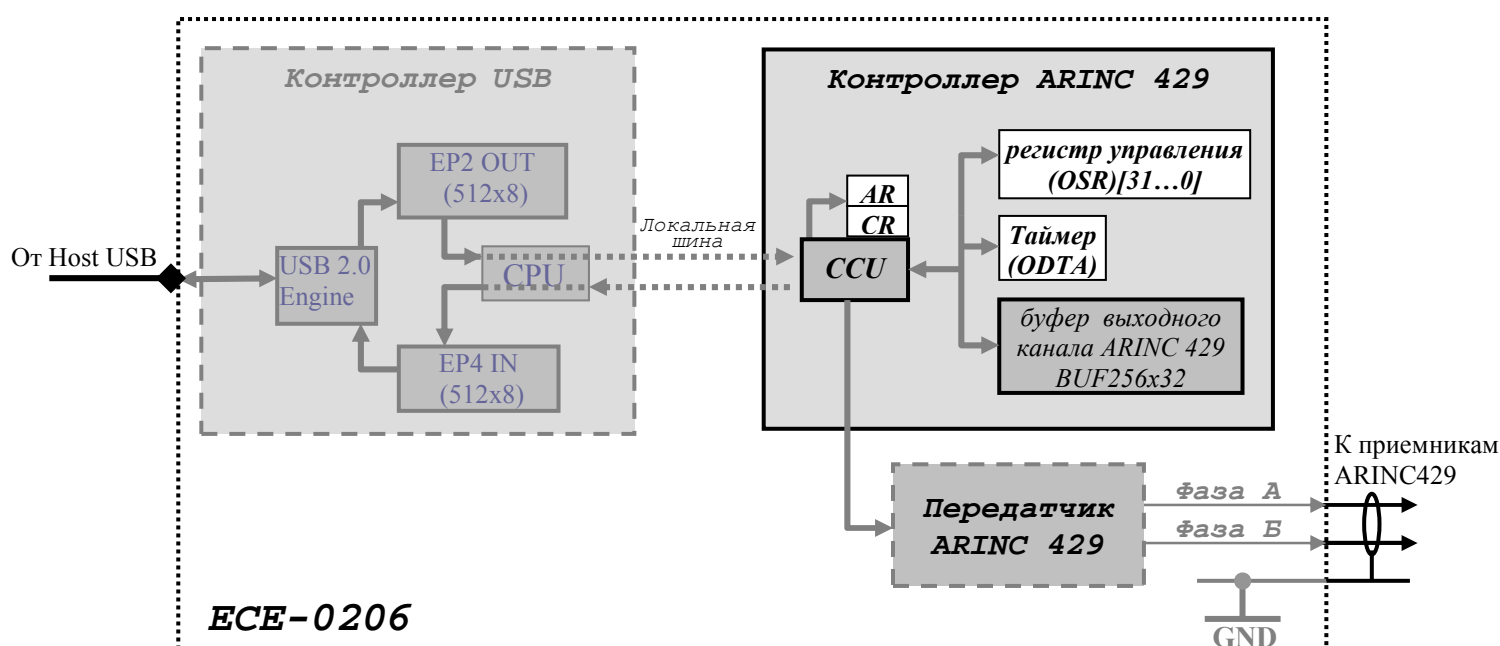
**Ch\_adr[3..0]** – адрес канала к которому будет запрос или обращение

= 0h – выходной канал 1

= 4h – входные каналы 1..4

## 9. Работа с выходным каналом

На приведенном ниже рисунке изображена структурная схема механизма передачи данных в выходной канал ARINC 429.



Контроллер ARINC429 в части выходного канала состоит из следующих элементов:

OSR (Output Setup Register) – регистр управления выходным каналом.

ODTA (Output Delay Timer) – таймер задержки выдачи [10,24мс...2,6112с].

BUF256x32 (Buffer 256x32bit) – буфер памяти на 1Кб с организацией 256x32бит.

Данные, которые необходимо передать в выходной канал формируются ПО на ПК пользователя, а затем передаются в буфер выходного канала BUF256x32. Затем производится настройка регистра управления и старт выдачи.

Рассмотрим эти 2 этапа:

## 10. Заполнение буфера BUF256x32.

Весь буфер представляет собой память на 256 32-х разрядных слов. Вся информация, записанная в буфер, доступна CCU для выдачи в канал ARINC 429. После задания режимов выдачи и старта канала, через установку регистра OSR, данные начинают поступать из BUF256x32 на передатчик и во внешнюю линию ARINC429.

В соответствии с форматом РТМ1495-75 буфер BUF256x32 можно представить следующим образом:

Номер слова буфера Выдачи в восьмеричной (шеснацатиричной) форме. (AR.Adr[7..0])	Номера битов 32-х разрядного слов ПК																															
	32	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
000 (00h)	Ч	МА	Данные(параметр)																			ИД	Адрес Параметра (АП)									
001 (01h)	Ч	МА	Данные(параметр)																			ИД	Адрес Параметра (АП)									
:																																
:																																
:																																
377 (FFh)	Ч	МА	Данные(параметр)																			ИД	Адрес Параметра (АП)									

**Ч** – Бит чётности.

**МА** – Матрицы «знак / состояние» (Описание функции матрицы и способы её кодирования см. РТМ 1495-75 п.2.1.3).

**ИД** – Поле идентификатора «источник / применение» (Назначение идентификатора, а также ограничения на его применение см. РТМ 1495-75 п.2.1.7).

Размер буфера обусловлен количеством всевозможных комбинаций значений Адреса Параметра:  $2^8 = 256$ .

Для заполнения буфера достаточно записать в конечную точку EP2 следующую последовательность:

**Adr Cmd Wrd1 Wrd2 ... Wrd<sub>n</sub>** где:

**Adr** = 00..FFh - адрес ячейки с которой начинается запись массива;

**Cmd** = «80h» - команда блочной записи массива

= «00h» - команда записи одного 32-х разрядного слова

**Wrd1 ... Wrd<sub>n</sub>** – 32-х разрядные слова данных, где  $1 \leq n \leq 256$ .

Из-за организации конечной точки EP2 как буфера размером 512байт заполнение всего буфера выходного канала BUF256x32 можно разбить на 3 итерации записи:

1. 00h 80h Wrd1 Wrd2 ... Wrd127

2. 7Fh 80h Wrd128 Wrd129 ... Wrd254

3. FEh 80h Wrd255 Wrd256

где **Wrd1-Wrd256** – данные пользователя.

Запись каждого слова занимает приблизительно 4мкс, т.е. после записи первого массива из 127 слов следующую запись можно производить не ранее чем через ~0.6мс.

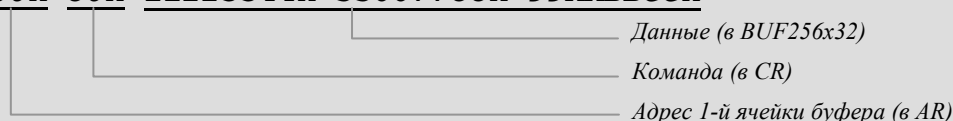
**Пример1:** Для заполнения, например, ячейки буфера выходного канала с адресом 09h, в конечную точку EP2 нужно записать следующую последовательность:

09h 00h 11223344h



**Пример2:** Для заполнения первых 3-х ячеек буфера выходного канала, в конечную точку EP2 нужно записать следующую последовательность:

00h 80h 11223344h 55667788h 99AABBCC h





## 11. Настройка режимов выходного канала и старт выдачи

Выходной канал допускает следующие режимы выдачи:

1. Однократная выдача массива;
2. Многократная выдача массива;
3. Циклическая выдача массива;

Для лучшего понимания режимов выходного канала представим временную модель выдачи данных в общем виде:

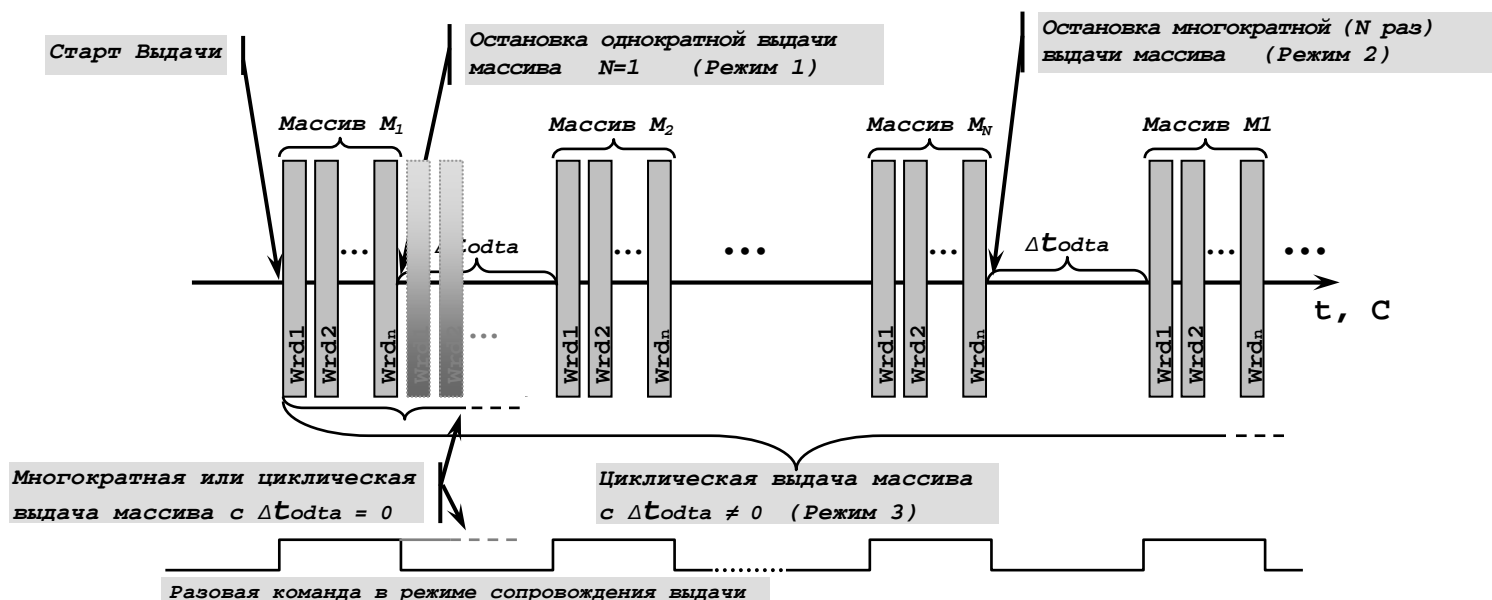


Рис. 1

**Массив** – это последовательность слов, выдаваемая в выходной канал из буфера BUF256x32. Чтение и выдача слов осуществляется последовательно, начиная с нулевой ячейки. Количество слов массива определяется пользователем и может принимать значение от 1 до 256 слов.

Выходной поток данных, как уже говорилось, формируется ССУ из буфера BUF256x32 в соответствии с настройками регистра OSR.

Рассмотрим **режим (1)**. После старта канала, контроллер ССУ начинает выдачу первого слова массива из нулевой ячейки буфера. Дальнейшая выборка из буфера происходит с инкрементом счетчика адреса до тех пор, пока его значение не достигнет размера массива заданного пользователем ( $OSR[16..9]=N_{wrd}$ ), после чего контроллер ССУ останавливает выдачу слов в канал.

**Режим (2)** отличается от режима (1) тем, что после выдачи одного массива контроллер ССУ не останавливает канал, а сравнивает текущий счетчик со значением количества выдаваемых массивов ( $OSR[24..17]=N_{array}$ ). В том случае если эти значения не совпали, контроллер ССУ запускает таймер ODTA на интервал  $\Delta t_{odta}$  заданный пользователем ( $OSR[32..25]=O_{dta}$ ). По истечению интервала  $\Delta t_{odta}$  выдача массива повторяется. Таким образом, цикл продолжается до момента совпадения текущего счетчика массивов со значением количества массивов, заданным пользователем ( $OSR[24..17]=N_{array}$ ). При этом условия канал останавливается до тех пор, пока пользователь снова не инициирует выдачу.

**Режим (3)** отличается от режима (2) только тем, что выдача массивов происходит циклически и непрерывно до момента остановки канала пользователем. Поскольку значение количества выдаваемых массивов для этого режима не имеет смысла, его значение задается  $N_{array} = 00h$ , и расценивается контроллером ССУ как признак циклического режима выдачи.

Рассмотрим настройку регистра OSR.

**Регистр управления выходным каналом OSR: (Доступ: запись/чтение).**

**OSR[32...25]** - интервал для таймера ODTA;

OdtA7	OdtA6	OdtA5	OdtA4	OdtA3	OdtA2	OdtA1	OdtA0
OSR32	OSR31	OSR30	OSR29	OSR28	OSR27	OSR26	OSR25

**OdtA(7..0)** – [00h..FFh] интервал для таймера ODTA.

Значение OdtA(7..0) задает временной интервал  $\Delta t_{odta}$  между выдачей массивов. При значении равно OdtA(7..0) = 00h выдача массивов происходит без паузы (т.е. с паузой 4T, предусмотренной стандартом).

Цена младшего разряда OSR25 составляет **10,24 мс**. Таким образом, максимально возможное значение интервала между массивами составит  $\Delta t_{odta} = 255 \times 10,24 \text{ мс} = 2,6112 \text{ с}$ .

**OSR[24...17]** - количество выдаваемых массивов;

Narray7	Narray6	Narray5	Narray4	Narray3	Narray2	Narray1	Narray0
OSR24	OSR23	OSR22	OSR21	OSR20	OSR19	OSR18	OSR17

**Narray(7..0)** – [00h..FFh] - количество выдаваемых массивов;

Кроме значения количества массивов ( $1_{dec} - 255_{dec}$ ), это поле регистра OSR имеет дополнительное свойство. Значение этого поля определяет тип выдачи: однократный либо многократный. При значении количества выдаваемых массивов Narray(7..0) = 00h контроллер CCU расценивает это как команду к циклической выдаче массива из BUF256x32. В случае если значение OSR(24..17) = Narray  $\neq$  00h = [01h..FFh], контроллер CCU произведет однократную (01h) или многократную (02h..ffh) выдачу Narray массивов.

**OSR[16...9]** - размер массива;

Nword7	Nword6	Nword5	Nword4	Nword3	Nword2	Nword1	Nword0
OSR16	OSR15	OSR14	OSR13	OSR12	OSR11	OSR10	OSR9

**Nword(7..0)** – [00h..FFh] размер массива.

Значение, Nword(7..0) определяет количество слов, которые будут составлять массив. Значения Nword(7..0) = 01h–FFh соответствуют количеству выдаваемых слов  $1_{dec} - 255_{dec}$ . Значение же Nword(7..0) = 00h соответствует количеству  $256_{dec}$  слов, т.е. всему буферу BUF256x32.

**OSR[8...1]** - настройки выходного канала;

Start/Stop Out	Err_en	DO_set1	DO_set0	ParityOut	33_31_bit	Freq1	Freq0
OSR8	OSR7	OSR6	OSR5	OSR4	OSR3	OSR2	OSR1



**Start/StopOut** – старт или остановка канала.

= «1» - при записи единицы в эту ячейку происходит старт канала выдачи.

= «0» - останов канал. Значение логического нуля в этой ячейке может быть записано как пользователем, так и контроллером ССУ при достижении условия останова канала.

В том случае если пользователь сформировал однократную/многократную выдачу, то момент остановки канала можно контролировать по состоянию бита OSR8. Для этого нужно читать регистр OSR с определенной периодичностью и проверять бит OSR8.Start/StopOut на равенство «0».

**Err\_en** – формирование ошибочного слова.

= «1» - в этом режиме контроллер ССУ формирует ошибочное, с точки зрения стандарта ARINC 429, слово данных. В этом режиме в соответствии со значением бита **OSR3.33\_31\_bit** контроллер ССУ добавит либо сократит количество бит в выдаваемом слове, т.е. от значения взятого из буфера BUF256x32 будет сформировано слово из 31 бита либо из 33 бит. Этот режим позволяет проверить принимающие системы на корректность обработки ошибок. (см. описание бита **OSR3.33\_31\_bit**)

= «0» - передатчик работает по стандарту ARINC429 (ГОСТ 18977-79) т.е. выдача 32-х битных последовательностей слов. В этом режиме значение бита **OSR3.33\_31\_bit** значения не имеет.

**DO\_set[1..0]** – биты управления выходной разовой команды.

Установкой битов **OSR6, OSR5** задается режим формирования выходной разовой команды **DO** (выход разъема X1.15):

= «00» - **DO(n-1)**, состояние выхода DO не изменится.

= «10» - **DO=Track**, в этом режиме выходные последовательности слов сопровождаются выдачей разовой команды DO (см. рис.1). Задержка между выдачей переднего фронта разовой команды и слова данных составляет 1T периода частоты канала.

= «01»\* - **DO=1**, на выход DO устанавливается активный уровень ТТЛ +5В относительно цифровой земли устройства.

= «11»\* - **DO=0**, на выход DO устанавливается уровень цифровой земли устройства 0В.

*\* при значении бита **DO\_set0 = '1'**, в байте записываемом в **OSR[8..1]**, состояние бита **Start/StopOut** значения не имеет, т.е после записи бит **Start/StopOut** в регистре **OSR[8..1]** не изменяется!*

**ParityOut** – формирование нечетности.

= «1» - при выдаче 32-х разрядных слов в 32-м разряде формируется бит нечетности. Значение 32-ого разряда из буфера BUF256x32 не учитывается.

= «0» - нечетность не формируется.

**33\_31\_bit\*** – формирование ошибки количества бит в выдаваемых словах.

= «1» - при записи единицы в эту ячейку количество бит в выдаваемом слове становится равным 33. В том случае если бит паритета установлен, то значение 33-его бита выдаваемого слова примет значение бита нечетности. В противном случае значение 33-его бита будет равняться «0».

= «0» - количество бит в выдаваемом слове становится равным 31. В том случае если бит паритета установлен, то значение 31-ого бита выдаваемого слова примет значение бита нечетности. В противном случае значение 31-ого бита будет соответствовать значению, записанному в буфер BUF256x32. Бит 32 игнорируется.

*\* значение бита **33\_31\_bit** учитывается только в том случае, если бит **Err\_en = «1»***

**Freq[1..0]** – частота выходного канала.  
= «00» - частота выдачи 12,5 КГц.  
= «01» - частота выдачи 50 КГц.  
= «1X» - частота выдачи 100 КГц.

**Пример3:** запись в регистр OSR:

00h 20h OSR[32..25] OSR[24..17] OSR[16..9] OSR[8..1]

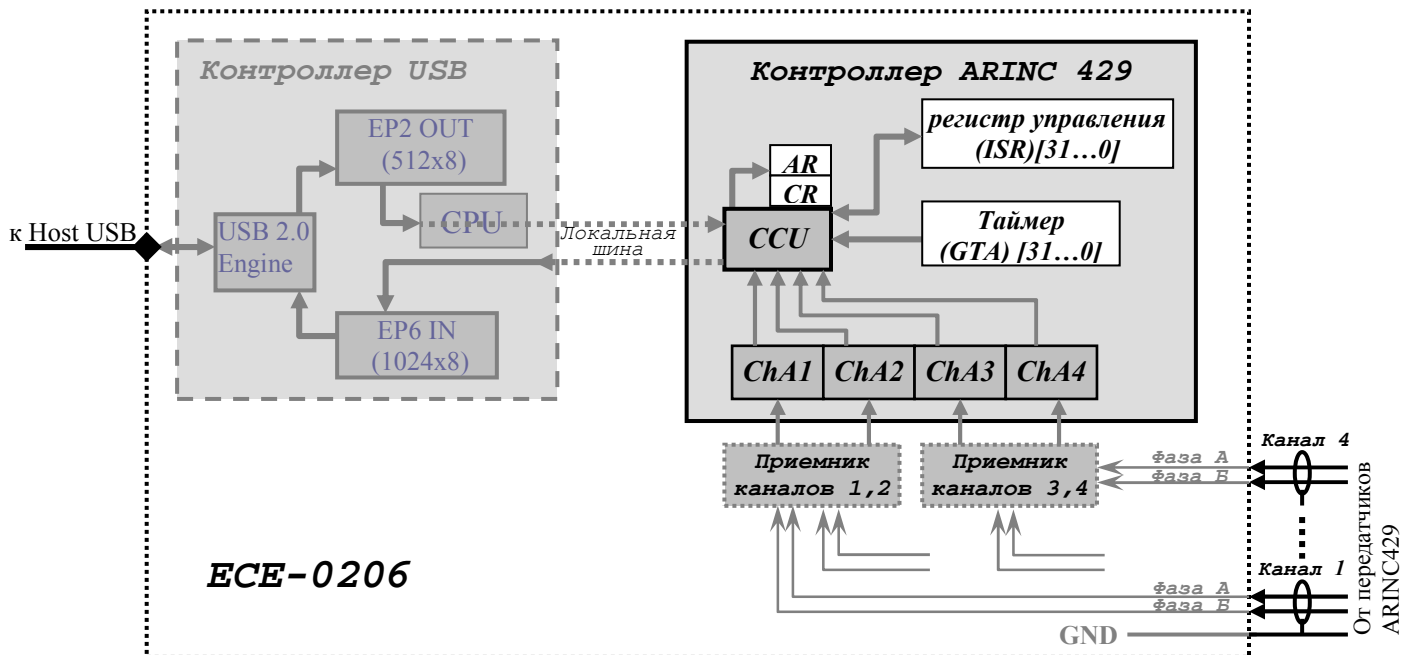
Формат записи регистра OSR допускает только полную модификацию всех 4-х байт регистра.

После старта канала существует возможность модификации регистра RSO «на лету». Каждое обновление регистра вступает в силу с момента полной выдачи очередного слова из буфера. Т.е. если изменение в регистре RSO произошло в момент выдачи, это никак не повлияет на текущее выдаваемое слово, а изменит только режим выдачи, начиная со следующего слова.

Аналогичным образом допускается модификация буфера выходного канала BUF256x32. В случае изменения буфера в процессе выдачи, и в том случае если выдача происходит из той же ячейки буфера, в которую пользователь записывает новое значение, модифицированное слово поступит в выходной канал только после завершения выдачи текущего (т.е. не модифицированного) слова.

## 12. Работа с входными каналами

На приведенном ниже рисунке изображена структурная схема механизма передачи данных от входных каналов ARINC 429 к хосту ПК пользователя.



Контроллер ARINC429 в части входных каналов состоит из следующих элементов:

ISR (Input Setup Register) – регистр управления входными каналами.

GTA (Global Timer) – глобальный таймер устройства до **4ч 46м** с точностью **4мкс**.

ChAx (Channel Analyzer) – каналный анализатор, где **x** – номер канала от 1 до 4.

Сигналы, приходящие по линиям ARINC 429, преобразуются приемниками, и поступают на соответствующие каналные анализаторы. В процессе поступления слов данных, каждый анализатор независимо проверяет логико-временную характеристику поступающего сигнала. Затем, на основании анализа, он сообщает контроллеру CCU о готовности данных принятого кода, а также об ошибках приема либо об их отсутствии. Контроллер, в свою очередь передает данные от анализатора в локальную шину, сопровождая их метками номера канала и времени момента приема слова (таймера GTA). Поскольку трафик от входных каналов относительно велик и требует опроса не реже фиксированного интервала времени, в контроллере USB для него выделена отдельная конечная точка EP6 IN типа Interrupt, размером 1024 байта с двойной буферизацией. Все данные, форматированные контроллером CCU, поступают по локальной шине напрямую в конечную точку EP6 IN, минуя CPU контроллера USB. Дальнейший механизм выборки данных зависит от программы управления хост контроллером на ПК пользователя.

*Значение Polling Interval (интервал опроса) для конечной точки EP6 IN задан = 01h. Это значение передается в дескрипторе конечной точки хосту, и является командой для выполнения опроса готовности буфера конечной точки EP6 IN с минимальным возможным интервалом. Этим обеспечивается возможность гарантированной доставки данных от входных каналов при максимальном трафике прослушиваемых систем ARINC429.*

Рассмотрим работу входных каналов на примере настроек регистра ISR.

**Регистр управления входными каналами ISR: (Доступ: запись).**

**ISR[32...25]** - настройки 1-ого канала;

Start/Stop IN1	--	--	TestMode1	ParityIN1	--	--	Fast/Slow1
ISR32	ISR31	ISR30	ISR29	ISR28	ISR27	ISR26	ISR25

**ISR[24...17]** - настройки 2-ого канала;

Start/Stop IN2	--	--	TestMode2	ParityIN2	--	--	Fast/Slow2
ISR24	ISR23	ISR22	ISR21	ISR20	ISR19	ISR18	ISR17

**ISR[16...9]** - настройки 3-ого канала;

Start/Stop IN3	--	--	TestMode3	ParityIN3	--	--	Fast/Slow3
ISR16	ISR15	ISR14	ISR13	ISR12	ISR11	ISR10	ISR9

**ISR[8...1]** - настройки 4-ого канала;

Start/Stop IN4	--	--	TestMode4	ParityIN4	--	Full/Short *	Fast/Slow4
ISR8	ISR7	ISR6	ISR5	ISR4	ISR3	ISR2	ISR1

**Start/StopINX** – старт или остановка входного канала X.

= «1» - при записи единицы в эту ячейку происходит старт соответствующего канала приема X=[1...4].

= «0» - при записи нуля, происходит остановка приема по соответствующему входу X.

**TestModeX** – тестовый режим входного канала X.

= «1» - установка тестового режима. В этом режиме выходной канал подключается к соответствующему (X=[1...4]) входному каналу внутри контроллера ARINC429, т.е. без участия приемника(ов). При этом выходной канал продолжает выдавать сигнал через передатчик в линию ARINC429.

Этот режим предназначен для тестирования исправности логической части интерфейса (контроллер USB + контроллер ARINC429), а также может применяться для отладки ПО пользователя и освоения работы с модулем при отсутствии передающих систем по ARINC429.

= «0» - основной режим. В этом режиме выходной канал подключен только к передатчику ARINC429 и формирует сигнал только во внешнюю линию.

\* Режим **Full/Short** описан в разделе 13: Режимы заполнения буфера конечной точки входных каналов EP6 IN.

**ParityINX** – контроль нечетности по входному каналу X.

= «1» - при записи единицы в эту ячейку включается механизм контроля нечетности поступающих слов по ARINC429. В случае ошибочного бита четности принятого слова в соответствующем поле байта сопровождения формируется код ошибки. (описание формата данных от входных каналов см. далее «Ошибки входных каналов»).

= «0» - поступающие слова по соответствующему каналу принимаются без контроля нечетности.

**Fast/SlowX** – установка частотного диапазона входного канала X.

= «1» - режим «Slow». В этом режиме устанавливается «медленный» диапазон контроля частоты поступающих слов по входному каналу X. В этом диапазоне безошибочно принимаются слова соответствующие формату ARINC429 и находящиеся в диапазоне частот **11–14,5 КГц**.

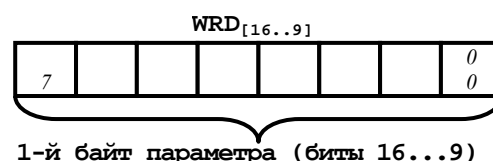
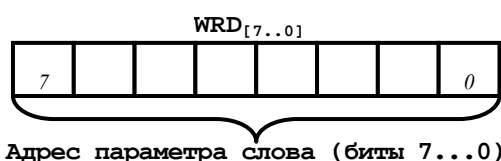
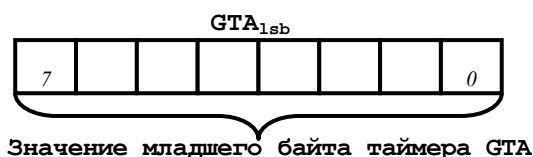
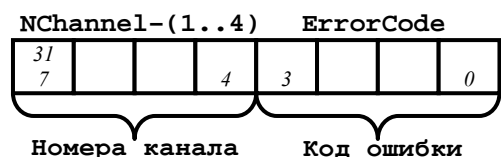
= «0» - режим «Fast». Это «быстрый» диапазон контроля частоты по входному каналу X. В этом диапазоне безошибочно будут приниматься слова в формате ARINC429 и находящиеся в диапазоне частот **36-101 КГц**.

### **13. Формат данных и алгоритм работы CCU в части входных каналов**

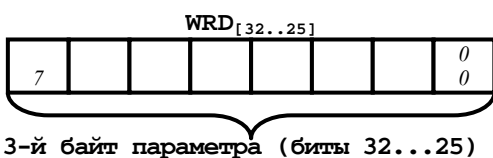
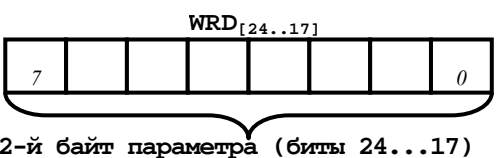
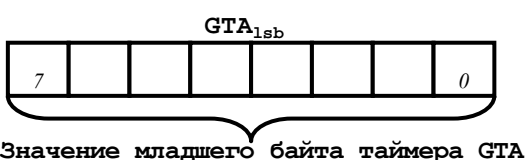
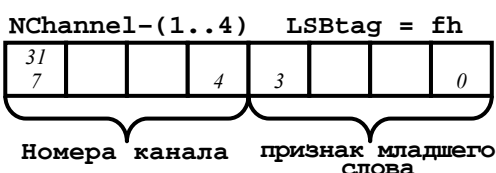
#### **13.1 Данные от входных каналов**

Все данные от входных каналов, как уже говорилось, форматируются контроллером CCU и передаются в конечную точку EP6 IN. Так как буфер конечной точки имеет архитектуру 1024x8 и является промежуточным хранилищем данных между контроллером ARINC429 и Хост контроллером на ПК пользователя, то и данные, поступающие в него должны быть форматированы таким образом, чтобы ПО могло распознать всю необходимую информацию. В связи с этим требованием в устройстве определен формат данных для буфера конечной точки EP6 IN от входных каналов. Любое принятое по каналу ARINC429 слово обрабатывается CCU и передается в буфер EP6 IN в виде 2-х 32-х разрядных слов MSW и LSW:

#### **Старшее слово MSW**



#### **Младшее слово LSW**



**NChannel** – поле в котором записывается номер канала от которого пришло слово.

**ErrorCode** – поле кода ошибки. В том случае если при приеме входного слова произошла известная ошибка, то код этой ошибке заносится в поле ErrorCode.

**GTA<sub>lsb</sub>** – значение младшего байта глобального таймера в микросекундах.

**LSBtag** – это 4-х битный признак младшего слова, всегда = Fh.

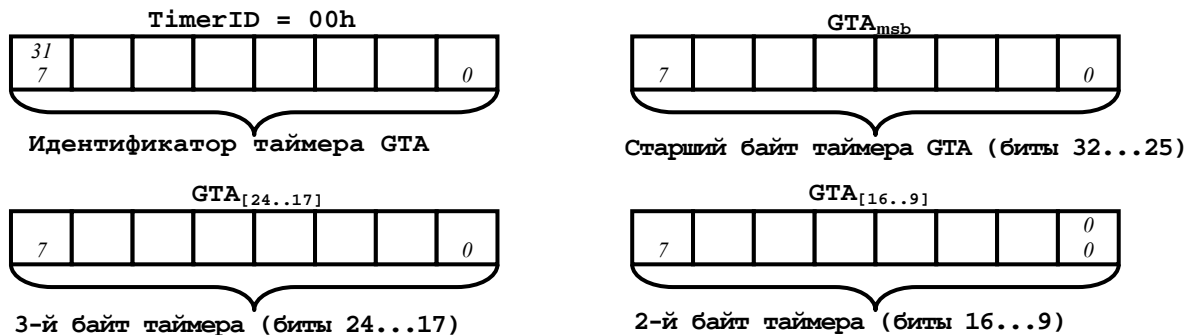
**WRD<sub>[32..25]</sub> WRD<sub>[24..17]</sub> WRD<sub>[16..9]</sub> WRD<sub>[8..1]</sub>** – 32-х разрядное слово данных от входного канала **NChannel**.

Таким образом, происходит заполнение буфера конечной точки EP6 IN данными от всех 4-х каналов. При этом между старшим MSW и младшим LSW словом от одного канала может попасть запись старшего и/или младшего слова от другого канала. Это может происходить в случае одновременного приема данных по двум каналам ARINC429. При этом значение второй половины, т.е. младшего слова, гарантированно запишется в буфер, но в ячейки со старшим адресом (либо попадет в следующий буфер).

### 13.2 Метки времени

В момент старта хотя бы одного канала (запись «1» в Start/StopINx регистра ISR) происходит запуск глобального таймера GTA. Таймер имеет 32 разряда и тактируется частотой 250 КГц (период счета 4 мкс). Для контроллера CCU таймер, как и любой канал, является источником данных и обрабатывается по готовности. Все время работы входных каналов разбито на промежутки в 1 мс (а точнее 1024 мкс), которые и являются сигналом готовности таймера GTA. При каждом отсчете таймером интервала  $\Delta t = 1024\text{мкс}$  контроллер CCU отправляет значения 3-х старших байт в буфер конечной точки EP6 IN. Таким образом, в буфер записываются метки времени с периодом в ~1 мс. Формат меток времени в буфере конечной точки представлен так:

#### Метка времени TLabel



**GTA<sub>msb</sub> GTA<sub>[24..17]</sub> GTA<sub>[16..9]</sub>** – это значение глобального таймера GTA.



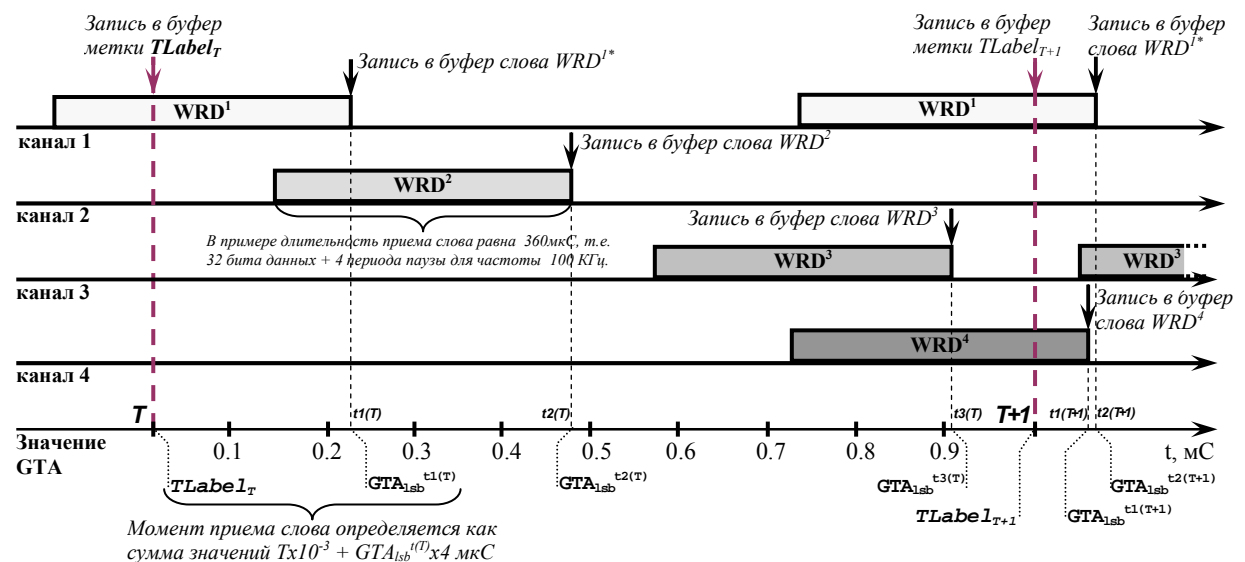


Таким образом, получается, что весь буфер размечен миллисекундными (точнее 1024 мкс) метками, между которыми расположены данные от каналов со значениями младшего байта таймера  $GTA_{1sb}^{t1(T)}$ , где  $t1(T)$  - это текущее время в микросекундах относительно метки  $TLabel_T$ . Для вычисления времени  $t1$  в микросекундах значение  $GTA_{1sb}^{t1(T)}$  нужно умножить на 4 т.е.:  $t1 = (GTA_{1sb}^{t1(T)} \times 4) \text{ мкс} = GTA_{[8..1]} \times 4 < 2$ . Сложив значение миллисекундной метки  $T$  и  $t1$  можно получить полное значение таймера для каждого слова, т.е. время прихода слова по каждому каналу относительно момента старта. Таким образом, получаем формулу для расчета значения момента прихода слова:

$$T_{\text{приема}} = (T + t1) = ((GTA_{[32..9]}^{(T)} \times 1,024) + ((GTA_{1sb}^{t1(T)} \times 4) / 1000)) \text{ мс.}$$

При этом значение  $GTA_{1sb}$  следует брать от старшей части слова MSW т.к. это более точное значение момента прихода данных. Значение  $GTA_{1sb}$  младшей половины слова LSW будет иметь значение с задержкой относительно MSW порядка 1-3 младших разрядов таймера (т.е. 4–12 мкс). Это связано с задержкой обслуживания канала контроллером ССУ. Формально, значение таймера  $GTA_{1sb}$  в младшей половине слова LSW вообще не нужно, но для того чтобы не усложнять формат протокола, это значение сохранено.

Восстановим временную картину во входном канале для рассмотренного примера фрагмента буфера:



Таким образом, видно, что структура данных в буфере конечной точки EP6 IN дает возможность восстановить не только информацию, но и временную картину поступления данных независимо по каждому входному каналу ARINC429.

Обратим внимание на возможные ситуации при записи данных в буфер EP6 IN. Одна из них, это не слитное размещение слова данных  $WRD^x$  в буфере. Во фрагменте буфера, рассмотренного выше, эта ситуация приведена на примере слова  $WRD^4$ . Рассмотрим этот пример: слово от 4-ого канала поступило в приемник, и установилась готовность контроллеру ССУ. Практически одновременно с ним, но чуть позже (порядка единиц микросекунд) поступила готовность от 1-ого канала. При этом контроллер ССУ уже успел передать старшую часть слова  $MSW_{4(T+1)}$  в буфер. Далее срабатывает механизм приоритетов каналов: канал с меньшим номером имеет высший приоритет. Контроллер переключается на передачу слова  $WRD^1$  от первого канала, а затем возвращается к обработке готовности от канала 4. Таким образом, в буфере между старшей и младшей частями слова от старшего канала могут быть записаны слова от младших каналов, которые поступили с небольшой задержкой относительно слова старшего канала. Исходя из этого, можно сделать вывод, что «склеивание» полного слова должно осуществляться

только по значению канала **NChannel** и определению младшей части слова по признаку **LSBtag**, без привязки к следованию данных в буфере.

Еще одна возможная ситуация размещения данных в буфере, это попадание метки времени **TLabel** между старшей и младшей частями слова **WRD<sup>x</sup>**. Таймер имеет наивысший приоритет и поэтому, как и в примере, описанном выше, может «разрывать» в буфере слова данных от каналов. В этом случае нужно применять аналогичный принцип «склеивания» по значению **NChannel** и **LSBtag**, а также обратить внимание на значение байта **GTA<sub>LSB</sub>** старшей половины слова **MSW**, который является точным указателем момента прихода слова.

#### **14 Режимы заполнения буфера конечной точки входных каналов EP6 IN**

В архитектуре последовательной шины USB применяется буферизация данных, и это свойство интерфейса вносит определенные задержки между моментом прихода слов данных и получением этих данных хостом ПК пользователя. В зависимости от трафика по входным каналам **ARINC429** скорость заполнения буфера конечной точки **EP6 IN**, а следовательно и задержка отправки буфера хосту, может быть различной. Для того чтобы изменять эту задержку в устройстве предусмотрено 2 режима заполнения буфера конечной точки **EP6 IN (1024x8)**:

**ISR[8...1]** - настройки 4-ого канала;

Start/Stop IN4	--	--	TestMode4	ParityIN4	--	Full/Short *	Fast/Slow4
ISR8	ISR7	ISR6	ISR5	ISR4	ISR3	ISR2	ISR1

**Full/Short** – режим заполнения буфера **EP6 IN**.

= «0» - режим готовности буфера при полном заполнении.

В этом режиме готовность буфера, а затем выборка данных хостом происходит только в момент полного заполнения, т.е. при заполнении буфера 1024-я байтами информации (данные от каналов + метки времени). Этот режим целесообразно применять в том случае если задержка между поступлением данных от каналов и получением их хостом (ПО пользователя) не имеет большого значения. В том случае если прием хотя бы по 1 каналу включен, а поступающих данных нет, т.е. буфер будет заполняться только метками времени, то период заполнения буфера составит: **(1024байт / 4байт)х1,024мс = 256х1,024 ≈ 262 мс**. При наличии данных во входном канале **ARINC429** это время будет только сокращаться обратно пропорционально количеству этих данных. Минимальное возможное время задержки при максимальном трафике по 4-м входным каналам на частоте 100 КГц составит **≈ 11 мс**.

= «1» - режим отправки буфера через фиксированный интервал **10,24 мс**.

В этом режиме готовность и отправка буфера происходит через фиксированный интервал времени 10 мс, не зависимо от того, какое количество данных в нем находится. Этот режим дает возможность получения оперативной информации от каналов о поступлении данных. Количество данных в этом случае будет зависеть от трафика по входным каналам **ARINC429**. В случае отсутствия поступающих данных по каналам, буфер будет представлять собой последовательность из 10 либо 11 меток времени (40 либо 44 байта), т.е. то количество меток, которое поступит за период 10,24 мс. Максимальное же количество данных от 4-х входных каналов на частоте 100 КГц составит приблизительно полный буфер **≈ 930-970 байт**.

## 15 Ошибки входных каналов (ErrorCode)

Каждый анализатор ChAx входного канала проверяет поступающий последовательный код на наличие ошибок формата ARINC429. В том случае если ошибка определяется, её код заносится в поле ErrorCode старшей части слова MSW. Как только анализатор определил ошибку в канале, он выставляет готовность и ждет обслуживания контроллером CCU. При этом в поле ErrorCode установлен код ошибки, а в полях данных остается значение слова (части слова, а точнее фактически принятых бит), при приеме которого возникла ошибка. Рассмотрим возможные ошибки и соответствующие им значения поля ErrorCode:

### Error 1 (ErrorCode = 8h)

Частота входного слова меньше заданной скорости приема.

В том случае если частота входного последовательного кода будет меньше нижней границы диапазона частот заданного в поле регистра **ISR.Fast/SlowX** для соответствующего канала, но не более чем в 4 раза меньше периода верхней границы диапазона частот, в поле ErrorCode запишется значение **<1000>b (8h)**. Для диапазона частот «Fast» (см. описание регистра ISR стр.12) ошибка Error1 возникнет при частоте входного сигнала в диапазоне  $25\text{КГц} < F_{\text{вх. сигнала}} < 36\text{КГц}$ , для режима «Slow» -  $3,6\text{КГц} < F_{\text{вх. сигнала}} < 11\text{КГц}$ .

### Error 2 (ErrorCode = 9h)

Количество бит в слове меньше 32-х, либо период сигнала больше 4Т для заданного диапазона частот.

В случае, если количество бит данных в слове меньше 32-х, либо частота входного последовательного кода для заданного диапазона частот **ISR.Fast/SlowX** меньше верхней границы диапазона более чем в 4 раза (т.е. период следования бит превышает минимальное значение паузы 4Т для заданного диапазона), в поле ErrorCode запишется значение **<1001>b (9h)**. Для диапазона «Fast» ошибка Error2 возникнет при частоте входного сигнала  $F_{\text{вх. сигнала}} \leq 25\text{КГц}$ , для диапазона «Slow» -  $F_{\text{вх. сигнала}} \leq 3,6\text{КГц}$ .

*Существует, также штатная ситуация, при которой возможно возникновение ошибки 9h. Это происходит в том случае, если модуль подключается к выходному каналу прослушиваемой системы в тот момент, когда в канале идет выдача слова. В этом случае, возможно, что на момент включения входного канала модуля в линии уже передается слово и прием начнется не с начала, а с произвольного бита слова. В этом случае анализатор ChAx воспримет обрывок слова как ошибочное слово с количеством бит меньше 32-х и сгенерирует ошибку 9h. В этом случае первые слова с ошибкой 9h можно просто игнорировать. Следующее слово уже будет принято правильно.*

### Error 3 (ErrorCode = Ah)

Количество бит в слове больше 32, либо период между словами меньше 4Т для заданного диапазона частот.

В том случае если количество бит данных в слове больше 32, либо интервал времени между предыдущим и следующим словом меньше 4Т для заданного диапазона частот, в поле ErrorCode запишется значение **<1010>b (Ah)**. Для диапазона «Fast» ошибка Error3 возникнет при интервале  $\Delta t_{\text{паузы}}$  между словами:  $\Delta t_{\text{паузы}} < 40\text{мкс}$ , для диапазона «Slow» -  $\Delta t_{\text{паузы}} < 276\text{мкс}$ .

#### **Error 4 (ErrorCode = Bh)**

Переополнение канального анализатора ChAx.

Это внутренняя ошибка, которая возникает в том случае если предыдущее слово не забрано из ChAx на момент прихода следующего, т.е. контроллер ССУ не успел обработать готовность канала на момент поступления следующего слова. Эта ситуация может произойти если канал работает на очень высокой частоте (не соответствующей стандарту ARINC429) либо высокочастотные помехи в канале были распознаны принимающей логикой как полезный сигнал. Такая ситуация возможна при частоте последовательного сигнала в канале более ~700 КГц. Значение в ErrorCode = <1011>b (Bh).

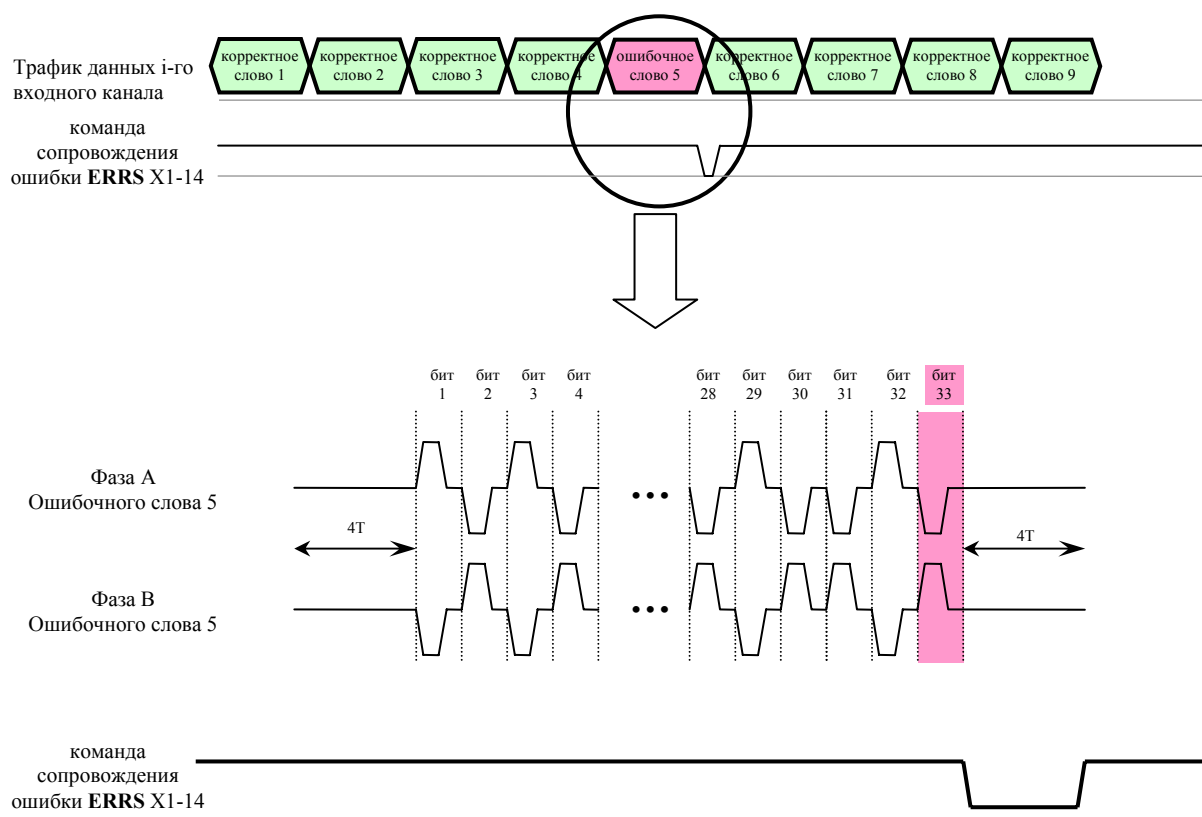
#### **Error 5 (ErrorCode = Ch)**

Ошибка контроля четности.

Ошибка возникает в случае, когда во входящем слове данных присутствует ошибочный бит четности, т.е. если слово не прошло проверку на нечетность. Ошибка может возникнуть только в том случае, если включен режим контроля нечетности канала (ISR.ParityINX = «1»). Значение в ErrorCode = <1100>b (Ch).

### **15.1 Команда сопровождения ошибки ERRS.**

В модуле предусмотрена возможность синхронизации осциллографа с моментом обнаружения анализатором ССУ ошибочного слова в канале. Опция позволяет зафиксировать на экране осциллографа развернутое слово в канале, которое было определено контроллером ССУ как ошибочное\* и записано в буфер канала с кодом ошибки (см. раздел 15). На рисунке представлен частный случай, иллюстрирующий работу режима синхронизации ошибки.

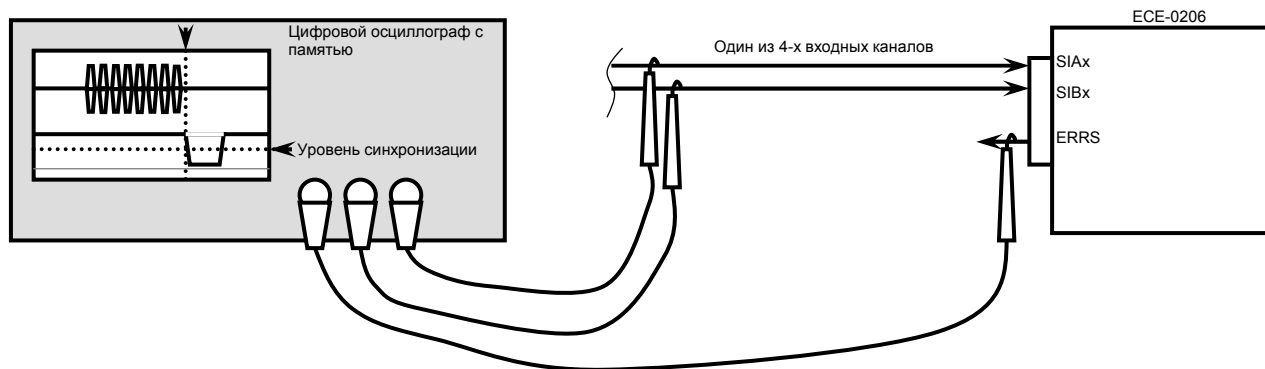


Линия синхронизации (выходная РК) находится на контакте 14 разъема X1. Тип выходной РК – «ключ на корпус», подключенная через 10 КОм к уровню питания +5В. Длительность импульса синхронизации порядка 30-40 мкс.

\* За ошибочное слово в канале контроллер ССУ может воспринять, в том числе и помеху, вызывающую срабатывание логики приемного каскада канала, и воспринятую как один бит слова (ошибка 9h). Помеха может вызывать также и ошибку Ah (см. раздел 15).

Стробы синхронизации формируются от каждого из 4-х входных каналов и объединенные по логическому «И» выдаются на управление выходным каскадом РК. Для синхронизации ошибки только по одному входному каналу необходимо программно остановить прием по остальным каналам либо физически отключить линии передачи от разъема. Однако, если пользователь уверен в отсутствии ошибок по остальным входным каналам эти действия можно не производить.

Ниже приведена типовая схема подключения осциллографа для контроля ошибочных слов в канале.

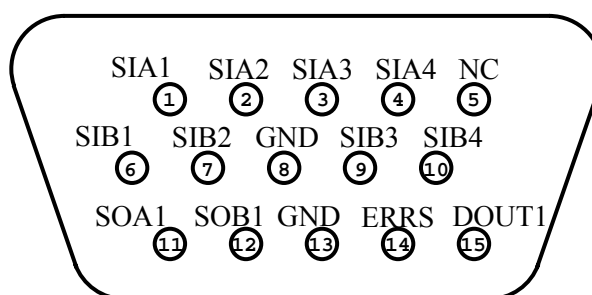


## 16. Цепи внешнего разъема X1

Внешний разъем X1 установлен на боковой панели корпуса и имеет тип DHS-15M (вилка).

Контакт	Цепь	Контакт	Цепь	Контакт	Цепь
1	SIA1	6	SIB1	11	SOA1
2	SIA2	7	SIB2	12	SOB1
3	SIA3	8	GND	13	GND
4	SIA4	9	SIB3	14	ERRS
5	NC	10	SIB4	15	DOUT1

Внешний вид разъема X1 DHS-15M



Где:

**SIAX** – (Serial Input phase **A**) – фазы A входного канала X (где X = 1...4);

**SIBX** – (Serial Input phase **B**) – фазы B входного канала X (где X = 1...4);

**SOA1** – (Serial Output phase **A**) – фазы A выходного канала 1 ;

**SOB1** – (Serial Output phase **B**) – фазы B выходного канала 1 ;

**DOUT1** – (**D**iscrete **O**utput) – выходная разовая команда;

**ERRS** – (**E**rror **S**ynchro **O**ut) – разовая команда сопровождения ошибки.