Радиомодуль портативного терминала Нуклон 5. Требования, описания, протоколы взаимодействия.

Версия 1.5

16-03-2016

# История изменений документа

|  |  |  |
| --- | --- | --- |
| **Дата** | **№ версии** | **Описание изменений** |
| 21.01.2016 | 1.0 | Начальное наполнение документа |
| 10.02.2016 | 1.1 | Добавлен раздел «Перечень команд GUI (STM) – CC1120».  Внесены уточняющие правки в раздел «Протокол обмена микроконтроллера STM32F071CBU6 c процессором NT1004»  Начал описывать алгоритм взаимодействия микросхем радиомодуля |
| 12.02.2016 | 1.2 | Добавлен раздел «Основные настроечные характеристики и параметры режимов взаимодействия» |
| 16.02.2016 | 1.2 | Добавлен рисунок с битовой конструкцией радиопротокола |
| 22.02.2016 | 1.3 | Изменено описание формата межпроцессорных сообщений (таблица 1) |
| 05.03.2016 | 1.4 | Внесены правки и дополнения:  -в протокол обмена между процессорами;  - в радиоинтерфейс. |
| 16.03.2016 | 1.5 | Внесены изменения в части:  - протокола обмена файловыми данными между терминальными устройства;  - протокола обмена между процессорами (а именно: изменен порядок бит в коде сообщения MSG, скорректировано описание некоторых типов сообщений в Таблице 1);  - радиоинтерфейса |
|  |  |  |

Протокол обмена микроконтроллера STM32F071CBU6 c процессором NT1004

Требования к протоколу обмена и интерфейсу взаимодействия радиомодуля с основной платой Нуклон 5.

Процессор радиомодуля STM32F071CBU6 должен быть подключен к центральному процессору NT1004 при помощи интерфейса UART.

Требования к GUI центрального процессора NT1004

GUI должна предусматривать возможность выбора режимов работы радиомодуля

- режим обмена голосом

- режим обмена данными (с подтверждением)

- режим обмена поправками RTK (без подтверждения)

Возможность ввода частоты приема и частоты передачи в диапазоне 410МГц – 480 МГц, с шагом кратным 25 кГц.

Возможность выбора режима усилителя мощности: номинальная или повышенная

Уровень приема RSSI приемника

Громкость динамика

Чувствительность микрофона

Возможность выбора источника данных (файла) в режиме передачи данных

Возможность отображения текущего состояния (прием, передача, дежурный прием)

Возможность отображения ошибки

Возможность отображения (по запросу) версии ПО микроконтроллера STM

Что еще добавить?

Как должен работать в режиме RTK?

Настройки, установленные c gui, должны храниться в энергонезависимой памяти радиомодуля (stm или fram). При включении (активации) радиомодуля должны передаваться на nt1004.

**Протокол обмена между процессорами**

Стандартный тип – UART.

Скорость обмена – 115200 бит/с

Ведущим устройством на шине является процессор NT1004, ведомым – микроконтроллер STM радиомодуля.

Функции, которые должен обеспечивать интерфейс

- усыплять/будить радиомодуль

- задавать режим функционирования радиомодуля

- управлять параметрами радиомодуля

- получать параметры от радиомодуля

- ?

Для реализации предполагается следующий логический интерфейс:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| FEND | LEN | MSG | ID | Data | CRC | FEND |

Основой протокола является протокол SLIP (UNIX™ Serial Link Interface Protocol), соответствующий стандарту RFC 1055. Передача данных осуществляется в двоичном виде, т.е. используются все возможные значения байта (00h…FFh). Для передачи служебной информации зарезервированы два кода: FEND = C0h (Frame End) и FESC = DBh (Frame Escape). Управляющий код FEND служит для обозначения начала и конца сообщения, а код FESC служит для передачи ESC-последовательностей. Если в потоке данных (в заголовке, в теле или в контрольной сумме сообщения) встречаются байты, значения которых совпадают с управляющими кодами, производится подмена этих байт ESC-последовательностями (механизм байт-стаффинга, byte stuffing). Код FEND заменяется последовательностью <FESC>, <TFEND>, а код FESC – последовательностью <FESC>, <TFESC>, где TFEND = DCh (Transposed FEND), TFESC = DDh (Transposed FESC). Коды TFEND и TFESC являются управляющими только в ESC-последовательностях, поэтому при передаче данных они в подмене не нуждаются.

LEN – 1 байт (размер данного и всех последующих полей дан без учета операции байт-стаффинга, которая может увеличить размер), размер поля Data, в байтах. Диапазон возможных значений LEN – от 0 до 128.

MSG – 1 байт, код сообщения.

Структура байта MSG:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Адрес | | | | Номер сообщения | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Адрес сообщения (4 бит):

0001b - микроконтроллер STM;

0010b - процессор NT1004.

Остальные значения зарезервированы для диагностики отдельных микросхем в светлом будущем.

Номер сообщения

Предназначен для последовательной нумерации отправляемых команд. Механизм нумерации сообщений позволяет идентифицировать сообщения и отслеживать их повторы. Если ведущее устройство не получило от ведомого устройства ответ на переданную команду, команда может быть повторена. При этом номер сообщения повторной команды должен быть таким же, как и номер исходной команды. Ответ от ведомого устройства должен иметь номер исходной команды.

ID – 1 байт, идентификатор сообщения

Предназначен для указания типа сообщения (команд, запросов, ответов). Перечень возможных идентификаторов указан в таблице 1.

Data – от 0 до 128 байт, тело сообщения

Размер и наполнение зависит от идентификатора сообщения (см. таблицу 1)

CRC – 1 байт, контрольная сумма сообщения

Контрольная сумма рассчитывается как XOR всех байт сообщения. Контрольная сумма рассчитывается перед операцией байт-стаффинга, проводимой над данными, для всего сообщения, исключая стартовые и стоповые байты FEND.

Перечень идентификаторов сообщений с соответствующими описаниями приведен в таблице:

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор сообщения, (название) | Описание сообщение | Размер данных | Комментарий |
| 00  (NOP) | Пустая команда | 1 байт | Данные произвольные |
| 01  (NOP\_BACK) | Ответ на пустую команду | 1 байт | Возвращает байт данных команды NOP |
| 02  (SET\_MODE) | Установить режим | 6 байт | 0 байт – код рабочего режима1;  1 байт – код настроек аудиопараметров2;  2-3 байт - код частоты передачи3;  4-5 байт - код частоты приема3 |
| 03  (SET\_MODE\_BACK) | Ответ на команду Установить режим | 1 байт | Код возврата, указывающий успешность выполнения команды установки режима |
| 04 (SEND\_DATA\_FRAME) | Передать фрейм данных | 1-128 байт | Передать данные на трансивер |
| 05 (SEND\_DATA\_FRAME\_BACK) | Ответ на команду  Передать фрейм данных | 1 байт | Код возврата, указывающий успешность выполнения команды передачи фрейма:  0 – фрейм не может быть передан (очередь на передачу переполнена, радиоканал занят или др. причина);  1 – фрейм успешно скопирован в очередь на передачу в радиоинтерфейс |
| 06 (TAKE\_DATA\_FRAME) | Принять фрейм | 1 байт | Принять данные с трансивера |
| 07 (TAKE\_DATA\_FRAME\_BACK) | Ответ на команду принять фрейм | 0-128 байт |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 08 (REQ\_CURRENT\_PARAM) | Запрос текущих параметров | 1 байт | Бит 0 – однократный/асинхронный ответ на запрос. В случае выбора асинхронного запроса ведомое устройство самостоятельно инициирует ответ с требуемыми параметрами в случае их изменения (например, уровень RSSI). Для выключения асинхронного режима ведущее устройство формирует повторный запрос точно такого же формата (?).  1-6 биты – маска запрашиваемых параметров:  бит 1 – запрос кода рабочего режима1;  бит 2 – запрос настроек аудиопараметров2;  бит 3 – запрос частоты передачи3;  бит 4 – запрос частоты приема3;  бит 5 – запрос значения RSSI;  бит 6 – запрос состояния радиоканала |
| 09 (REQ\_CURRENT\_PARAM\_BACK) | Ответ на запрос текущих параметров | 2-9 байт (в зависимости от запроса) | 0 байт – повторяет байт запроса команды REQ\_CURRENT\_PARAM (указывает тип запроса и маску возвращаемых параметров);  1 байт – код рабочего режима1;  2 байт – код настроек аудиопараметров2;  3-4 байты – код частоты передачи3;  5-6 байты – код частоты приема3;  7 байт – значение RSSI;  8 байт – состояние радиоканала:  1h – активный прием;  2h – дежурный прием;  3h – передача |
| 0A (SOFT\_VER) | Запрос версии ПО STM | 0 байт |  |
| 0B (SOFT\_VER\_BACK) | Ответ на запрос версии ПО STM | 2 байта | Значение текущей версии ПО  0 байт – major-версия;  1 байт – minor-версия  Формат: (major-версия).(minor-версия) |
|  |  |  |  |

1 – код рабочего режима:

0-1 биты:

01b – голос

10b – данные с подтверждением (обмен произвольными файлами);

11b – данные без подтверждения (обмен поправками RTK);

бит 2 – резерв;

бит 3 – мощность передатчика: малая/полная;

бит 4 – спать/не спать;

5-6 биты – резерв Скорость обмена в режимах обмена данными

00b – «режим 4.8»

01b – «режим 9.6»

10b – «режим 19.2»

11b – «режим 48»

Описания характеристик каждого режима приведено далее по тексту.

Внимание

Скорость обмена в режимах обмена данными должна устанавливаться только в режимах обмена данными. В режиме обмена голосом всегда включен «режим 4.8»

2 – код настроек аудиопараметров:

0-2 биты – усиление звукового выхода;

3-5 биты – усиление звукового входа;

6-7 биты – резерв

3 - Код частоты – порядковый номер частотного канала из диапазона 410-480 МГц с канальным шагом 25 кГц.

**Протокол обмена информационными данными между терминалами:**

В режиме обмена произвольными файлами терминалы должны использовать следующий протокол взаимодействия.

Первый фрейм данных, который передается с помощью команды SEND\_DATA\_FRAME, должен содержать специальный заголовок, указывающий основные атрибуты передаваемого файла: имя, размер.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Фрейм 1 | Фрейм 2 | Фрейм 3 | … | Фрейм N |
| Заголовок | Данные файла | Данные файла | … | Данные файла |

Формат заголовка:

|  |  |  |
| --- | --- | --- |
| DataType | FileName | FileSize |
| 2 байта | 32 байт | 4 байта |

DataType – тип данных:

0x0001 – файл

FileName – имя файла (включая расширение, пример: “abcde.txt”)

FileSize – размер файла, байт

Полезных данных при этом в первом фрейме содержаться не должно (только служебные).

Следующие за первым фреймы в поле data должны состоять только из полезных данных. Для того, чтобы точно идентифицировать длину пакетов, в интерфейсе взаимодействия по радиоканалу, должно быть предусмотрено поле, указывающее количество полезных байт в пакете

**Перечень команд GUI (STM) – CC1120**

В данном разделе приведены логические команды и функции, необходимые для работы с трансивером СС1120 при управлении с GUI.

Стандартный тип – UART.

Скорость обмена – до 115200 бит/с

Общая логика выполнения команды должны быть следующей:

- при выборе типа сообщения из GUI на STM должен отправляться код сообщения

- STM должен распознавать принятый код сообщения и формировать последовательность управляющих воздействий, соответствующих идентификатору сообщения

Такие же команды и функции должны выполняться STM при управлении трансивером CC1120 без GUI (при управлении с платы с процессором NT1004).

Логический протокол взаимодействия GUI-stm и NT1004 – stm в конечном итоге должен быть идентичным (таким, как описан в разделе «Протокол обмена микроконтроллера STM32F071CBU6 c процессором NT1004»)

GUI должен содержать элементы, позволяющие настраивать:

- микросхему трансивера СС1120 (повторять GUI разработанное для проекта «Моноблок»);

- микросхему вокодера CMX7262;

- комплексные команды вида «передача», «прием», «калибровка» и т.п.;

Перечень команд приведен в таблице 2

Таблица 2

|  |  |
| --- | --- |
| Наименование команды | Описание |
| Transiever\_Verify()  Проверка трансивера | Запрашивает версию ПО трансивера |
| TX\_FIFO\_Control()  Проверка FIFO Tx | Проверяет количество байт, содержащихся в FIFO Tx |
| RX\_FIFO\_Control()  Проверка FIFO Rx | Проверяет количество байт, содержащихся в FIFO Rx |
| Manual\_calibration()  Калибровка трансивера | Запускает команду калибровки трансивера |
| Reset()  Сброс | Запускает команду сброса трансивера |
| Tx\_FIFO\_flush()  Проверка FIFO Tx | Очищает содержимое FIFO Tx |
| Rx\_FIFO\_flush()  Проверка FIFO Rx | Очищает содержимое FIFO Rx |
| TX()  Режим Передача | Запускает команду передачи трансивера |
| Status()  Статус | Запрашивает содержимое статусного байта STATUS |
| IDLE() | Запускает команду перехода в режим IDLE() |
| MARC() | Запрашивает текущее состояние трансивера в активном режиме |
| SFSTXON() | Запускает команду калибровки синтезатора трансивера |
| LQI\_VAL() | Запрос байта характеризующего качество связи по результатам принятого пакета |
| RX() | Запускает команду прием |
| Rx\_FIFO\_read() | Чтение буфера FIFO\_Rx |
| Tx\_FIFO\_write() | Запись буфера FIFO\_Tx |
| Записать конфигурацию  CHOSE\_CONF | Выбрать и загрузить конфигурацию трансивера |
| Прочитать конфигурацию  READ\_CONF | Прочитать текущую конфигурацию |
| Записать частоту  WRITE\_FREQ | Записать частоту трансивера |
| Прочитать частоту  READ\_FREQ | Прочитать частоту трансивера |
| Записать регистр  WRITE\_REG | Записать в указанный регистр значение |
| Прочитать регистр  READ\_REG | Прочитать из указанного регистра значение |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Нужно предусмотреть в GUI возможность выполнения сообщений , предусмотренных «Протоколом обмена микроконтроллера STM32F071CBU6 c процессором NT1004» (см таблицу 1)

- 00 (NOP) Пустая команда;

- 01 (NOP\_BACK);

- 02 (SET\_MODE) Установить режим;

- 03 (SET\_MODE\_BACK);

- 04 (SEND\_DATA\_FRAME) Передать фрейм;

- 05 (SEND\_DATA\_FRAME\_BACK);

- 06 (TAKE\_DATA\_FRAME) Принять фрейм;

- 07 (TAKE\_DATA\_FRAME\_BACK);

- 08 (REQ\_CURRENT\_PARAM) Запрос текущих параметров;

- 09 (REQ\_CURRENT\_PARAM\_BACK);

- 0A (SOFT\_VER) Запрос версии ПО;

- 0B (SOFT\_VER\_BACK).

**Основные настроечные характеристики и параметры режимов взаимодействия**

Канальная скорость передачи голосовых данных (FEC вокодера включен) 3.6 Кбит/с

Вокодер может пакетировать и передавать данные по 1,2,3,4 фрейма. Один фрейм – это 20 мс голоса или 9 байт данных с FEC кодированием.

Всего по радиоканалу требуется передавать пакеты длительностью 180 мс или 81 байт данных. Для этого STM должен накопить эти данные, полученные в 3 пакетах от вокодера по 3 фрейма в каждом.

Еще 2 байт стоит зарезервировать под служебную информацию.

Микроконтроллер должен иметь возможность конфигурирования радиомодуля в четырех режимах:

- режим 4.8. Режим предполагает возможность обмениваться голосом и данными с канальной скоростью 4.8 кБит/с

- режим 9.6. Режим предполагает возможность обмениваться голосом и данными с канальной скоростью 9.6 кБит/с

- режим 19.2 Режим предполагает возможность обмениваться голосом и данными с канальной скоростью 19.2 кБит/с

- режим 48. Режим предполагает возможность обмениваться голосом и данными с канальной скоростью 48 кБит/с

Общий вид пакета, передаваемого по радиоканалу, при обмене голосом, представлен на рисунке

Общий вид пакета, передаваемого по радиоканалу, при обмене голосом, представлен на рисунке

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Преамбула | Синхропосылка | Данные служебные | Данные полезные | CRC |

Преамбула – 12 байт;

Синхропосылка – 4 байт;

Данные служебные – 5 байт.

Формат служебных данных:

typedef struct {

uint8\_t dstAddress; // адрес получателя

uint8\_t srcAddress; // адрес источника

uint8\_t dataType; // тип данных (речь / гарант. данные / негарант. данные)

uint8\_t dataSize; // размер полезных данных в сообщении, байт

uint8\_t reserve;

} structRadioMsgHeader;

0й байт – адрес получателя (00h – широковещательная рассылка (всем) или индивидуальный адрес)

1й байт – адрес источника (передающее устройство должно помещать сюда собственный адрес)

2й байт – тип передаваемых данных:

01h – речевые данные (режим обмена голосом);

02h – данные с подтверждением (режим обмена данными);

03h – данные без подтверждения (режим обмена данными RTK);

3й байт – размер полезных данных в текущем пакете, байт

4й байт – резерв.

CRC – 2 байт.

Преамбула, синхрослово, CRC – добавляются трансивером автоматически, не учитываются при программировании и настройке payload, однако должны учитываться при расчете общей скорости передачи данных в канале.

Длина пакета (packet length) = служебные данные + полезные данные =5+81 = 86 байт

Для передачи голосового пакета длиной 180 мс на скорости 4800 бит/с требуется 108 байт. Такой должна быть общая длина пакета.

Общая длина пакета: 12+4+5+81+2 = 104 байта.

Сводная таблица с базовыми параметрами конфигурации для режима 1 («Голосовой режим 1») приведена ниже. Параметры режимов подобраны таким образом, чтобы индекс модуляции был близок к 0.5

Режим 4.8

|  |  |
| --- | --- |
| Название параметров | Значение параметра |
| **СС1120** |  |
| Символьная скорость  Symbol rate | 4800 бит/с |
| Канальная скорость  Bit rate | 4800 бит/с |
| Девиация  Deviation | 1200 Гц |
| Вид модуляции  Modulation format | 2-GFSK |
| Манчестер-код  Manchester enable | выкл |
| Полоса цифрового фильтра приемника | 10 кГц |
| Мощность передатчика | 15 дБмВт |
| Режим трансивера | High Perfomance |
| PA ramping | вкл |
| Режим длины пакета  Packet length mode | Фиксированная длина, 87 байт (1 адрес+5 служ + 81 голос) |
| Whitening | вкл |
| Режим обмена | пакетный |
| Проверка адреса  Device address | Включена, при этом 00 и FF широковещательные |
| Длина преамбулы | 12 байт |
| Длина синхрослова | 4 байт |
| Байт адреса | вкл |
| CRC, LQI | вкл |
| Обмен данными с микроконтроллером | По прерыванию gpio2.4 |
|  |  |
| **CMX7262** |  |
| Режим | Encode mode при передаче /Decode моде при приеме |
| FEC кодер/декодер | Включен, режим «hard decision» |
| Noise Gate | Выключен  Порог срабатывания  Время анализа |
| Пакетирование (количество фремов в пакете) | Включено (3) |
| Обмен данными с микроконтроллером | По прерыванию IRQN.57 |
|  |  |

Режим 9.6

|  |  |
| --- | --- |
| Название параметров | Значение параметра |
| **СС1120** |  |
| Символьная скорость  Symbol rate | 9600 бит/с |
| Канальная скорость  Bit rate | 9600 бит/с |
| Девиация  Deviation | 2400 Гц |
| Вид модуляции  Modulation format | 2-GFSK |
| Манчестер-код  Manchester enable | выкл |
| Полоса цифрового фильтра приемника | 20 кГц |
| Мощность передатчика | 15 дБмВт |
| Режим трансивера | High Perfomance |
| PA ramping | вкл |
| Режим длины пакета  Packet length mode | Фиксированная длина, 87 байт (1 адрес+5 служ + 81 данные) |
| Whitening | выкл |
| Режим обмена | пакетный |
| Проверка адреса  Device address | Включена, при этом 00 и FF широковещательные |
| Длина преамбулы | 12 байт |
| Длина синхрослова | 4 байт |
| Байт адреса | вкл |
| CRC, LQI | вкл |
| Обмен данными с микроконтроллером | По прерыванию IRQN.57 |
|  |  |

Режим 19.2

|  |  |
| --- | --- |
| Название параметров | Значение параметра |
| **СС1120** |  |
| Символьная скорость  Symbol rate | 19200 бит/с |
| Канальная скорость  Bit rate | 19200 бит/с |
| Девиация  Deviation | 4800 Гц |
| Вид модуляции  Modulation format | 2-GFSK |
| Манчестер-код  Manchester enable | выкл |
| Полоса цифрового фильтра приемника | 40 кГц |
| Мощность передатчика | 15 дБмВт |
| Режим трансивера | High Perfomance |
| PA ramping | вкл |
| Режим длины пакета  Packet length mode | Фиксированная длина, 87 байт (1 адрес+5 служ + 81 данные) |
| Whitening | выкл |
| Режим обмена | пакетный |
| Проверка адреса  Device address | Включена, при этом 00 и FF широковещательные |
| Длина преамбулы | 12 байт |
| Длина синхрослова | 4 байт |
| Байт адреса | вкл |
| CRC, LQI | вкл |
| Обмен данными с микроконтроллером | По прерыванию gpio2.4 |
|  |  |

Режим 48

|  |  |
| --- | --- |
| Название параметров | Значение параметра |
| **СС1120** |  |
| Символьная скорость  Symbol rate | 24000 бит/с |
| Канальная скорость  Bit rate | 48000 бит/с |
| Девиация  Deviation | 12000 Гц |
| Вид модуляции  Modulation format | 4-GFSK |
| Манчестер-код  Manchester enable | выкл |
| Полоса цифрового фильтра приемника | 50 кГц |
| Мощность передатчика | 15 дБмВт |
| Режим трансивера | High Perfomance |
| PA ramping | вкл |
| Режим длины пакета  Packet length mode | Фиксированная длина, 87 байт (1 адрес+5 служ + 81 данные) |
| Whitening | вкл |
| Режим обмена | пакетный |
| Проверка адреса  Device address | Включена, при этом 00 и FF широковещательные |
| Длина преамбулы | 12 байт |
| Длина синхрослова | 4 байт |
| CRC, LQI | вкл |
| Обмен данными с микроконтроллером | По прерыванию gpio2.4 |
|  |  |

–––––––––––––––– трэш –––––––––––––––––––––––––-трэш––––––––––––––––––––––––––-

Следующие за первым фреймы в поле data должны содержать заголовок, состоящий из ЧЧ байт, содержащий

- признак пакета с подтверждением или без

- признак первого пакета

- признак последнего пакета

- количество байт, добавленных в конце к полезным данным до полного заполнения пакета

- номер текущего пакета 3 байта

**Алгоритм взаимодействия аппаратного обеспечения микросхем радиомодуля**

Управляющим модулем на плате радиомодуля является микроконтроллер STM 32F071C.

Определены следующие интерфейсы взаимодействия:

- Внешний для взаимодействия с основной платой с процессором NT1004, или для взаимодействия с GUI при настройке и регулировке – UART+GPIO;

- внутренний для взаимодействия STM c вокодером CMX7262 – SPI1 (CBUS) (CS=SPI\_NSS\_2);

- внутренний для взаимодействия STM c ЦАПом AD5601 – SPI1 (CS = SPI\_NSS)

- внутренний для взаимодействия микроконтроллера STM c трансивером СС1120 – SPI2+4GPIO (конфигурация GPIO будет определена при программировании).

- внутренний для взаимодействия STM c FRAM FM24W256-G – I2C.

*Нам сейчас этот fram нужен вообще?*

Описание порядка взаимодействия микросхем:

**Передача голоса**

Режим предварительно устанавливается в gui.

При выборе голосового режима в gui NT1004 должен «разбудить» процессор STM радиомодуля по UART

Размер передаваемых трансивером полезных (голосовых) данных за один пакет должен составлять 180 мс (эквивалентно 81 байтам полезной информации в каждом пакете). Таким образом, STM должен осуществлять накопление пакетов от вокодера (нужно накапливать 9 фреймов по 20 мс). Возможно, нужно ввести дополнительный 1 байт для служебной информации

Начальное состояние:

- вокодер в idle режиме,

- трансивер в idle режиме.

- stm ожидает прерывания на ноге PA8.29 от тангенты, ожидает прерывания от трансивера на ноге gpio2.4 на сс1120 (или PB2.20 на STM). Примечание – прерывание от PTT имеет высший приоритет, если оно пришло, следует остановить прием и перейти на передачу.

По факту готовности данных от вокодера он может выставлять прерывание по IRQN.57 на PC14.3 STM.

Рабочее состояние

По приходу прерывания от PTT:

- разбудить вокодер, разбудить трансивер

- сообщить NT1004 об установке режима «передача»

- запросить параметры, которые настроены в gui

- установить смещение на УМ

- начать накапливать данные, приходящие с вокодера

- передавать фрейм данных на трансивер для передачи

- ждать прерывания от трансивера о готовности принять данные

- обрабатывать прерывание

- проверять состояние нажатой кнопки PTT

- по факту отпускания кнопки PTT очистить буферы

- сообщить NT1004 об установке режима «дежурный прием»

- перевести микросхемы трансивера и вокодера в idle

**Передача данных**

Режим предварительно устанавливается в gui и отправляется на STM в команде «Установить режим» «SET\_MODE»

Прием

Начальное состояние

Перывание по gpio о наличии полезных данных в буфере fifo