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

Версия 1.1

09-02-2016

# 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 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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

- режим обмена поправками RTK

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?

**Протокол обмена между процессорами**

Стандартный тип – UART.

Скорость обмена – до 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……………….data | CRC | FEND |

FEND = C0. 1 байт-признак начала обмена

FEND = C0 1 байт-признак конца сообщения

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.

<FESC> = DBh

LEN – 1 байт содержащий значение количества байт данных поля data. Количество байт данных не должно превышать 128 байт

MSG – 1 байт, код сообщения.

Структура байта MSG:

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

Адрес сообщения (4 бит)

«01» - микроконтроллер STM;

«10» - процессор NT1004.

Остальные значения зарезервированы для диагностики отдельных микросхем в светлом будущем.

Номер сообщения

Предназначен для последовательного нумерации отправляемых сообщений. Используется для избежания коллизий в случае перезапроса отправляемых сообщений.

ID – 1 байт, идентификатор сообщения

Предназначен для указания типа сообщения (команд, запросов, ответов) с указанием параметров сообщения.

CRC - Размер поля контрольной суммы сообщения: 8 бит.

XOR всех байт сообщения (за исключением старт - байта и стоп-байта)

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 байты данных команды NOP |
| 02  () | Установить режим | 4 байт | 1 байт  Два бита:  01 – голос  10 – данные  11 – данные RTK  3й бит - резерв  4й бит – мощность малая/полная  5й бит – спать/не спать  06-07 резерв  2 байт  3 бита громкость  3 бита чувствительность микрофона  2 бита - резерв  3,4 байт  Значение частоты, начиная с 410 МГц, заканчивая 480 МГц кратно 25 кГц. |
| 03  () | Ответ на команду Установить режим | 1 байт | Возвращает байт данных |
| 04 () | Передать фрейм | 128 байт | Передать данные на трансивер |
| 05 () | Ответ на команду  Передать фрейм | 1 байт |  |
| 06 () | Принять фрейм | 1 байт | Принять данные с трансивера |
| 07 () | Ответ на команду принять фрейм | 128 байт |  |
| 08 () | Запрос текущих параметров | 1 байт | 0й бит – 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 (?).  1-3 бит – код (порядковый номер) байта для возврата  000 все байты – см. сообщение 09  001 первый байт - см. сообщение 09  010 второй байт см сообщение 09  011 первый и второй байты – см. сообщение 09  100 третий и четвертый байт (текущая частота) – см. сообщение 09  101 пятый байт (значение RSSI) – см сообщение 09  110 шестой байт (2 бита (статус PTT, ошибки) – см сообщение 09  111 – пятый и шестой байт – см сообщение 09 |
| 09 () | Ответ на запрос текущих параметров | 2-7 байт (в зависимости от запроса) | 1й байт – повторяет байт запроса 08  2й байт (такой же как в сообщении «установить режим»):  Два бита:  01 – голос  10 – данные  11 – данные RTK  3й бит - резерв  4й бит – мощность малая/полная  5й бит – спать/не спать  06-07 резерв  3й байт (такой же как в сообщении «установить режим»)  3 бита громкость  3 бита чувствительность микрофона  2 бита – резерв  4,5 байт  Значение частоты  6й байт – значение RSSI  7й байт  2 бита (статус PTT)  01 прием  10 дежурный прием  11 передача  6 бит – код ошибки  0 – ошибки не зафиксировано  Остальные биты зарезервированы |
| 0A () | Запрос версии ПО | 0 байт |  |
| 0B () | Ответ на запрос версии ПО | 1 байт | Значение текущей версии ПО |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 должно 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\_Vreifity()  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  COSE\_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 команд , предусмотренных «Протоколом обмена микроконтроллера STM32F071CBU6 c процессором NT1004» (см таблицу 1)

- Пустая команда;

- Установить режим;

- Передать фрейм;

- Принять фрейм;

- Запрос текущих параметров;

- Запрос версии ПО;

- и ответы на них.

**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 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 36 байт.

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

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**

Прием

Начальное состояние

Перывание по gpio о наличии полезных данных в буфере fifo

Режим: голосовой обмен:

Основные характеристики

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

Вокодер передает пакеты по 36 байт

Микроконтроллер должен иметь возможность конфигурирования радиомодуля в четырех режимах:

Сводная таблица с базовыми параметрами конфигурации для режима 2 приведена ниже

|  |  |  |  |
| --- | --- | --- | --- |
| Название параметров | Значение параметра | Название регистров | Значение регистров |
| Частота несущей  Carrier frequency |  |  |  |
| Символьная скорость  Symbol rate |  |  |  |
| Канальная скорость  Bit rate |  |  |  |
| Девиация  Deviation |  |  |  |
| Вид модуляции  Modulation format |  |  |  |
| Манчестер-код  Manchester enable |  |  |  |
| Полоса цифрового фильтра приемника |  |  |  |
| Мощность передатчика |  |  |  |
| Режим трансивера |  |  |  |
| PA ramping |  |  |  |
| Режим длины пакета  Packet length mode |  |  |  |
| Whitening |  |  |  |
| Режим обмена |  |  |  |
| Проверка адреса  Device address |  |  |  |