

УТВЕРЖДЕН
RU.CГМА.00044-01 13 01-ЛУ

БЕЗОПАСНЫЙ ЛОКОМОТИВНЫЙ ОБЪЕДИНЕННЫЙ КОМПЛЕКС
МАСШТАБИРУЕМЫЙ БЛОК-М
Программное обеспечение
СН/БЛОК
Описание программы
RU.CГМА.00044-01 13 01
Листов 17

Инд. № прогн	Подп. и дата	Взам. инд. №	Инд. № дийл	Подп. и дата

АННОТАЦИЯ

В настоящем документе представлены условия применения, назначение и логическая структура программного обеспечения модуля съёмного носителя информации СН.

СОДЕРЖАНИЕ

Аннотация	2
1 Общие сведения.....	4
2 Функциональное назначение	5
3 Описание логической структуры.....	6
4 Используемые технические средства.....	10
5 Вызов и загрузка.....	11
6 Входные и выходные данные.....	12
Приложение 1 (Справочное) Перечень принятых сокращений	13
Приложение 2 (Обязательное) Сборка программного модуля СН/БЛОК	13
Лист регистрации изменений.....	17

1 ОБЩИЕ СВЕДЕНИЯ

Наименование программы – Программное обеспечение модуля съемного носителя информации безопасного локомотивного объединенного комплекса – масштабируемого БЛОК-М (далее по тексту – ПО СН/БЛОК).

Обозначение программы – RU.CГМА.00044-01.

Язык программирования – ARM Си

ПО СН/БЛОК предназначено для работы в составе комплекса БЛОК-М или иных локомотивных систем для регистрации информации. Программа записана во внутреннюю память микроконтроллера типа STM32F101VC. Программное окружение для работы ПО СН/БЛОК не требуется.

2 ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

ПО СН/БЛОК реализует следующие функции:

- реализация обмена данными с устройствами БС-СН и УСК по последовательному беспроводному каналу;
- сохранение полученных данных в энергонезависимой памяти типа NAND;
- чтение и передача записанных ранее данных из NAND-памяти считывающему устройству;
- управление памятью (записанными данными) для обеспечения безотказной работы;
- поддержка расширенных протоколов обмена при работе с БС-СН/САУТ и БС-СН/БЛОК-02, БС-СН/БЛОК-03;
- загрузчик позволяет производить обновление программного обеспечения Рабочего модуля с помощью специального сервисного устройства.

Перечень принятых сокращений приведён в приложении 1.

Порядок сборки исполняемого модуля ПО СН/БЛОК приведён в приложении 2.

3 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

Проект включает в себя 23 файла:

Имя файла	Назначение файла
1 Main.c	Главный модуль программы, включающий функцию main() и ряд вспомогательных функций
2 Main.h	Заголовочный файл, включающий определение глобальных Переменных и пользовательских типов, используемых в программе
3 proj_def.h	Заголовочный файл проекта с определением глобальных типов
4 ConfigPh.c	Функции инициализации периферийных устройств контроллера STM32F101xxx : RCC, GPIO, UART, NVIC
5 Init_proc.c	Функции инициализации пользовательских структур данных при запуске программы
6 Operation.c	Модуль с описанием функций обработки сообщений интерфейса обмена данными с БС-СН (УСК)
7 Mem_menager.c	Модуль с описанием некоторых функций работы с внешней энергонезависимой памятью
8 Oper_mem.c	Базовые функции работы с внешней NAND-памятью
9 Oper_mem.h	Заголовочный файл с описанием функций
10 Timr_menager.c	Модуль с описанием функций для работы с внутренними таймерами микроконтроллера
11 Nand_mem2.h	Заголовочный файл с описанием типов и функций для работы с памятью NAND
12 Nand_mem2.c	Содержит описание аппаратно-зависимых функций низкого уровня для доступа к внешней NAND-памяти: чтение страницы, запись страницы, очистка блока памяти и др
13 Stm32f10x_it.c	Функции обработчиков прерываний
14 Stm32f10x_it.h	Заголовочный файл
15 Stm32f10x_conf.h	Конфигурационный файл библиотеки
16 Chk_data.c	Функции расчета CRC-16
17 Console.c	Отладочные функции для вывода в консоль
18 debug_fun.c	Отладочные функции.

Имя файла	Назначение файла
19 protocol.h	Заголовочный файл.
20 Protocol_1a.c	Функции обработки расширенного протокола обмена
21 Protocol_2.c	Функции обработки протокола обмена САУТ
22 remap.h	Заголовочный файл
23 remap2.c	Дополнительные функции менеджера памяти по обработке сбойных блоков памяти

3.1 Общие положения

Программное обеспечение ПО СН/БЛОК состоит из двух независимых исполняемых модулей: модуль Загрузчика и Рабочий модуль.

В проекте Рабочего модуля использованы функции библиотеки периферийных устройств процессора STM32F10x (STM32F10x Standard Peripherals Library Drivers) – STM32F10x_StdPeriph_Driver v. 3.4.0 от 10/15/2010 (Copyright 2010 STMicroelectronics). Список используемых модулей библиотеки приведен в файле stm32f10x_conf.h .

Для инициализации ядра микроконтроллера задействованы функции библиотеки Cortex Microcontroller Software Interface Standard (CMSIS) Version 1.30 - 30 October 2009 (Copyright © ARM Ltd).

В проекте модуля Загрузчика аналогичным образом используются библиотеки STM32F10x_StdPeriph_Driver и CMSIS, указанные ранее.

В структуре программы Рабочего модуля можно выделить два блока: секция инициализации и основной цикл программы.

Блок инициализации на первом этапе выполняет настройку аппаратных средств микроконтроллера: ARM-ядра и периферийных узлов, - для выполнения поставленных задач. Рабочая частота ядра устанавливается равной 36,0 МГц. Включается тактирование задействованных периферийных устройств. Активируются USART, таймеры, контроллер прерываний, порты ввода/вывода,

встроенный контроллер внешней памяти (NAND), интерфейс SPI и внешняя вспомогательная память NVRAM. Путем тестового обращения к микросхемам внешней памяти определяется возможность дальнейшей работы программы и блока СН в целом.

На втором этапе производится инициализация (установка начальных значений) внутренних структур для работы "менеджера памяти" путем считывания конфигурационной информации, хранимой в энергонезависимой памяти носителя.

3.2 Основной цикл

Работа программы в основном цикле строится "вокруг" обработки приемного буфера сообщений. Сообщения (пакеты данных) поступают на приёмник USART с выхода приемника беспроводного индуктивного канала, и принимаются программой асинхронно, по прерываниям.

После получения очередного сообщения определяется целостность и корректность полученных данных путем расчета контрольной суммы пакета, предусмотренной протоколом обмена.

В зависимости от типа полученной команды: запись данных, чтение данных, дополнительные команды – вызываются функции, реализующие необходимый функционал.

Вне зависимости от типа полученного пакета формируется ответ, и через USART отправляется получателю.

Аварийный выход из цикла, с последующим перезапуском контроллера, возможен только при отказе внешней памяти, о чем информирует менеджер памяти.

3.3 Функции менеджера памяти

Основной задачей менеджера памяти является управление всем объемом памяти ЗУ, размещенной на микросхеме NAND-памяти, и создание нескольких

независимых потоков на запись/чтение данных. При этом физическое размещение данных по тем или иным адресам ЗУ остается в компетенции программы СН.

Весь объем ЗУ (256 Мбайт) в условиях завода-изготовителя разделяется на несколько областей: системную, область данных и резервную область. Область данных, в свою очередь, разделена для использования (регистрации данных) системами БЛОК/КЛУБ, САУТ, КПД-3 и другими. В пределах выделенного раздела менеджер памяти размещает (сохраняет) принимаемые данные.

При этом реализуются следующие функциональные особенности:

- запись "по кольцу", когда вновь поступающие данные записываются в ЗУ независимо от наличия там свободного места, при этом затираются наиболее старые данные;
- данные размещаются постранично (используется особенности организации NAND-памяти);
- начало каждой записи (каждого фрагмента) - определяется в момент подключения блока СН - располагается в начале страницы;
- данные в ЗУ и от ЗУ передаются постранично, объемом 512 байт;
- "прозрачная" буферизация данных при записи: содержание буфера модифицируется при поступлении очередных данных, изменения сохраняются в FRAM-памяти;
- каждая операция записи контролируется программно-аппаратным методом путем запроса статуса микросхемы.

4 ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

ПО СН/БЛОК работает в составе ПО БЛОК-М и иных локомотивных систем.

Используемый микроконтроллер - STM32F101VC (ЦП – 36 МГц, ОЗУ – 32 Кбайт, ПЗУ – 256 Кбайт).

5 ВЫЗОВ И ЗАГРУЗКА

Программа начинает автоматически выполняться при подаче электропитания на съёмный носитель СН.

Управление сначала получает Загрузчик, затем спустя 1,5 секунды, управление передается Рабочему модулю.

6 ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

Входными данными для ПО СН/БЛОК являются:

- конфигурация разделов ЗУ, считываемая из системной области;
- сообщения, поступающие по последовательному беспроводному каналу от БС-СН или УСК.

Выходными данными для ПО СН/БЛОК являются ответные сообщения, формируемые по запросам БС-СН или УСК.

ПРИЛОЖЕНИЕ 1

(Справочное)

ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ

БЛОК-М – безопасный локомотивный объединенный комплекс масштабируемый;

БС-СН – блок связи со съёмным носителем;

ЗУ – запоминающее устройство;

ПО – программное обеспечение;

СН/БЛОК – съёмный носитель информации;

УСК – устройство считывания кассет регистрации.

ПРИЛОЖЕНИЕ 2

(Обязательное)

СБОРКА ПРОГРАММНОГО МОДУЛЯ СН/БЛОК

1 Цель сборки

В результате сборки будет получен исполняемый файл программного модуля СН версии «26» подверсии «7.1» с наименованием «Cass_v26-7-1(44).bin», длиной 98 304 байт, и контрольной суммой CRC32 равной «C3CC2285», предназначенный для исполнения на микроконтроллере STM32F101VCT6 фирмы STMicroelectronics.

2 Необходимые технические средства

Для сборки исполняемого файла программного модуля центрального вычислителя необходимы технические средства со следующими минимальными характеристиками: ПЭВМ с частотой центрального процессора 2,5 ГГц, объемом оперативной памяти 2 Гбайт, объемом жесткого диска 50 Гбайт.

3 Необходимые программные средства

Сборка исполняемого файла программного модуля МП выполняется в среде операционной системы «Windows XP Professional SP3» (32-разрядная версия), путем применения среды разработки «IAR Embedded Workbench for ARM» версии 6.21.1.2846, с встроенным компилятором «IAR C/C++ Compiler for ARM» версии 6.21.1.52794, формирующим выходной код исполняемого файла из команд целевого микроконтроллера STM32F101.

4 Подготовка к сборке

Из источника хранения эталонных файлов необходимо, соблюдая структуру вложенности каталогов, скопировать папку, содержащую файлы исходных текстов (с расширениями .c, .h), файлы проекта среды разработки (с расширениями .eww, .dtp, .ewd, .ewp, .dep), и конфигурационные и

вспомогательные файлы (с расширениями .icf, .s) в каталог «D:\Progs\ST32» на сборочном стенде, состоящем из необходимых для сборки технических средств и содержащем необходимые программные средства. Изменить имя скопированной папки на «Proj_Cass2_v26». Структура папки проекта должна содержать папку «inc» с описаниями глобальных типов и переменных.

Дополнительно в каталоге «D:\Progs» создать папку «Libraries», в которую следует установить библиотеку Cortex Microcontroller Software Interface Standard (CMSIS) версии 1.3 (30.10.2009). А также в каталоге «D:\Progs\ST32» создать папку «Libraries», в которую следует установить библиотеку STM32F10x Standard Peripherals Library Drivers (StdPeriph_Driver) версии 3.4.0 (15.10.2010).

5 Выполнение сборки

Запустить среду программирования «IAR Embedded Workbench for ARM» версии 6.21.1.2846.

Выбрать пункт меню «File» в левом верхнем углу и далее «Open» и «Workspace». Появится окно для выбора проекта, в котором необходимо выбрать папку с проектом и открыть файл «ws_cas.eww». После этого в течение некоторого времени произойдет загрузка проекта (рисунок 2.1).

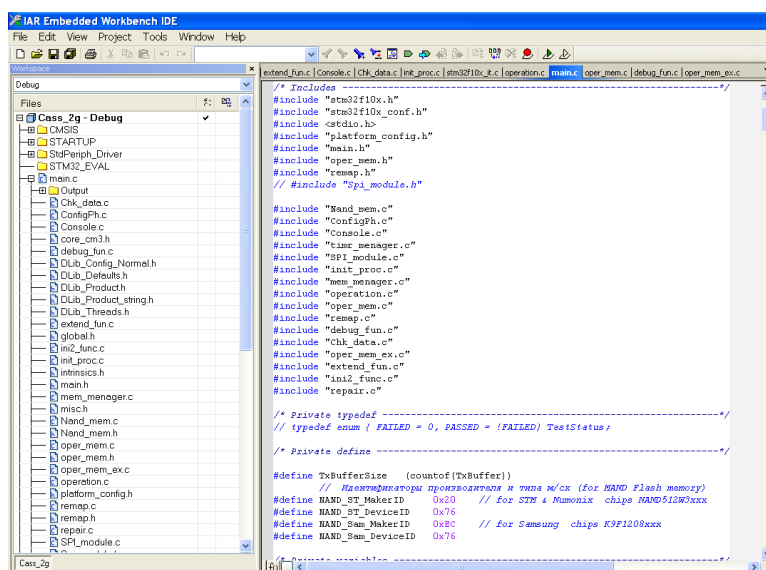


Рисунок 2.1 - Рабочее окно проекта среды программирования «IAR Embedded Workbench»

В настройках проекта (команда «Option» меню «Project») выбрать вкладку «Linker», где в окне «Linker configuration file editor» (рисунок 2.2), установить/проверить значения следующих параметров: `intvec start = 0x0800300`, `ROM Start = 0x0800300`, `ROM End = 0x080FFFFFF`, `RAM Start = 0x0200000`, `RAM End = 0x02017FFF`, `CSTACK = 0x400`, `HEAP = 0x400`.

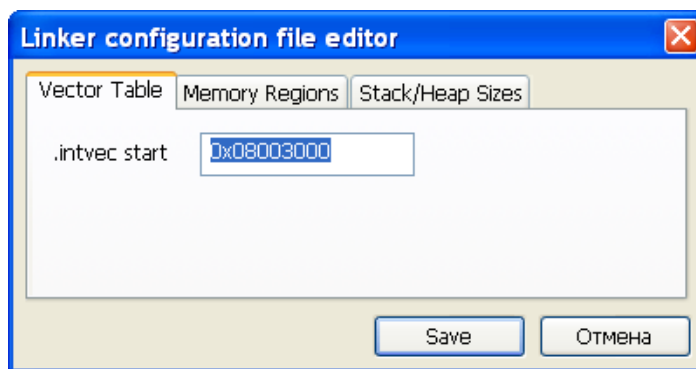


Рисунок 2.2 - Рабочее окно настроек компоновщика

После этого запустить компиляцию проекта путем выбора меню «Project» и команды «Rebuild all».

6 Получение исполняемого файла

При отсутствии ошибок в исходных текстах в результате сборки будет получен бинарный файл «Cass_v26.bin», который будет записан в папку «Debug\Exe» внутри папки проекта.

Для получения файла с наименованием «Cass_v26-7-1(44).bin» следует переименовать созданный бинарный файл в Проводнике Windows.

[illegible]