

БЕЗОПАСНЫЙ ЛОКОМОТИВНЫЙ ОБЪЕДИНЕННЫЙ КОМПЛЕКС –
МАСШТАБИРУЕМЫЙ БЛОК-М

Программное обеспечение

Программный модуль ввода и вывода дискретных сигналов

Описание программы

RU.СГМА.00042 13

Листов 21

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дудл.	Подп. и дата

АННОТАЦИЯ

В настоящем документе представлены условия применения, назначение и логическая структура программного обеспечения модуля ввода и вывода дискретных сигналов.

СОДЕРЖАНИЕ

Аннотация	2
1 Общие сведения.....	4
2 Функциональное назначение	5
3 Описание логической структуры.....	6
4 Используемые технические средства.....	14
5 Вызов и загрузка.....	15
6 Входные и выходные данные.....	Ошибка! Закладка не определена.
Приложение 1 (Справочное) Перечень принятых сокращений	16
Приложение 2 (Обязательное) Сборка программного модуля ввода и вывода дискретных сигналов.....	17
Лист регистрации изменений.....	Ошибка! Закладка не определена.

1 ОБЩИЕ СВЕДЕНИЯ

Наименование программы – Программное обеспечение модуля ввода и вывода дискретных сигналов безопасного локомотивного объединенного комплекса – масштабируемого БЛОК-М (далее по тексту – ПО ВВД).

Обозначение программы - RU.СГМА.00042.

Язык программирования – Си.

ПО ВВД предназначено для работы в составе комплекса БЛОК-М и иных локомотивных систем. Программа записана во внутреннюю память микроконтроллера типа STM32F205VC. Используется компилятор IAR ANSI C/C++ Compiler V7.50.2.10312/W32 for ARM. Для связывания объектных модулей используется редактор связей IAR ANSI C/C++ Compiler V7.50.2.10312/W32 for ARM фирмы IAR Systems. Программное окружение для работы ПО ВВД не требуется.

2 ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Программа блока ВВД реализует следующие функции:

- в соответствии с принимаемыми из CAN-линии сообщениями выполняет включение клапана КОН, ЭПВ №266, формирование сигнала "Разбор тяги" и управление пневматическими клапанами ТВ и ОВ;
- принимает дискретные сигналы, поступающие на вход блока, а именно: состояние ключа ЭПК, сигнала тревожной кнопки, сигнала положения контроллера, Тифон, Свисток и др. – и передает в линию CAN сообщения с текущим состоянием входных сигналов;
- выполняет опрос датчиков давления и формирует CAN-сообщения с текущими значениями давления в тормозной магистрали.

Перечень принятых сокращений приведён в приложении 1.

Порядок сборки исполняемых модулей ПО ВВД приведён в приложении 2.

3 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

Проект включает в себя 107 файлов:

Имя файла	Назначение
1 byteUtils.c	Утилиты для установки/сброса битов в байте
2 ByteUtils.h	Утилиты для установки/сброса битов в байте
3 CAN.c	Драйвер CAN
4 CAN.h	Драйвер CAN
5 CAN_FIFO.c	Очередь сообщений CAN
6 CAN_FIFO.h	Очередь сообщений CAN
7 CanDescriptors.h	Определения дескрипторов CAN-сообщений
8 CANMessageHandler.c	Обработчик сообщений CAN
9 CANMessageHandler.h	Обработчик сообщений CAN
10 CANPROG.c	Модуль обновления прошивки по CAN
11 CANPROG.h	Модуль обновления прошивки по CAN
12 ConfigNumberSender.c	Модуль, периодически отправляющий текущее конфигурационное число функциональности модуля вывода дискретного
13 ConfigNumberSender.h	Модуль, периодически отправляющий текущее конфигурационное число функциональности модуля вывода дискретного
14 Configurator.c	Модуль, извлекающий значения параметров из конфигурационного числа
15 Configurator.h	Модуль, извлекающий значения параметров из конфигурационного числа
16 DynamicConfig.c	Реализация алгоритма ввода ячейки в конфигурацию ЦО
17 DynamicConfig.h	Реализация алгоритма ввода ячейки в конфигурацию ЦО

Имя файла	Назначение
18 LED.c	Модуль управления светодиодной индикацией блока ВВД
19 LED.h	Модуль управления светодиодной индикацией блока ВВД
20 main.c	Основной программный модуль
21 SoftVersionSender.c	Модуль, отправляющий сообщение с номером версии ПО
22 SoftVersionSender.h	Модуль, отправляющий сообщение с номером версии ПО
23 types.h	Определение типов данных
24 VirtualTimer.c	Программный таймер
25 VirtualTimer.h	Программный таймер
26 BrakeController.c	Модуль управления пневматическим торможением по командам от САУТ
27 BrakeController.h	Модуль управления пневматическим торможением по командам от САУТ
28 BrakeControllerCommon.h	Определения, общие для модулей управления пневматическим торможением и ЭПТ
29 BrakeControllerElectroPneumatic.c	Модуль управления электропневматическим торможением (ЭПТ) по командам от САУТ
30 BrakeControllerElectroPneumatic.h	Модуль управления электропневматическим торможением (ЭПТ) по командам от САУТ
31 BrakeControllerElectroPneumaticPort.c	Определение платформозависимых функций, используемых модулем управления ЭПТ BrakeController.c
32 BrakeControllerElectroPneumaticPort.h	Определение платформозависимых функций, используемых модулем управления ЭПТ BrakeController.c
33 BrakeControllerPort.c	Определение платформозависимых функций, используемых модулем управления пневматическим торможением BrakeController.c

Имя файла	Назначение
34 BrakeControllerPort.h	Определение платформозависимых функций, используемых модулем управления пневматическим торможением BrakeController.c
35 DiscreteOutputs.c	Модуль, управляющий дискретными выходами
36 DiscreteOutputs.h	Модуль, управляющий дискретными выходами
37 DoorLock.c	Модуль, управляющий блокировкой дверей электропоезда по значению текущей скорости
38 DoorLock.h	Модуль, управляющий блокировкой дверей электропоезда по значению текущей скорости
39 EPK_toggle.c	Модуль, управляющий ключом разрыва петли экстренного торможения
40 EPK_toggle.h	Модуль, управляющий ключом разрыва петли экстренного торможения
41 EPV266.c	Модуль, управляющий ключом ЭПВ266
42 EPV266.h	Модуль, управляющий ключом ЭПВ266
43 FailSafeOutputsConfiguration.h	Безопасное состояние ключей дискретного вывода
44 KON.c	Модуль, управляющий ключом КОН
45 KON.h	Модуль, управляющий ключом КОН
46 mvd.c	Основной модуль, выполняющий функции ячейки "Вывод"
47 mvd.h	Основной модуль, выполняющий функции ячейки "Вывод"
48 mvd_SysDiagnostics.c	Модуль, формирующий сообщение SYS_DIAGNOSTICS ячейки Вывод
49 mvd_SysDiagnostics.h	Модуль, формирующий сообщение SYS_DIAGNOSTICS ячейки Вывод

Имя файла	Назначение
50 PET_break.c	Модуль, управляющий ключом разрыва петли экстренного торможения
51 PET_break.h	Модуль, управляющий ключом разрыва петли экстренного торможения
52 pressures.c	Модуль, хранящий значения давлений, полученных от датчиков ДДИ
53 pressures.h	Модуль, хранящий значения давлений, полученных от датчиков ДДИ
54 PressureSender.c	Модуль, отправляющий в CAN значения давлений в тормозах
55 PressureSender.h	Модуль, отправляющий в CAN значения давлений в тормозах
56 SautTractionCutoff.c	Модуль, управляющий ключом разбора тяги
57 SautTractionCutoff.h	Модуль, управляющий ключом разбора тяги
58 ConfigurableFeatures.h	Описание конфигурируемых функций модуля "Вывод"
59 KeyIndexes.h	Определение индексов ключей дискретного вывода
60 mvdConfigTable.h	Описание параметров модуля дискретного вывода, определяемых конфигурационным числом от МПХ
61 mvdTimings.h	Настройки временных интервалов для модуля дискретного вывода
62 analogSignal.h	Структура "Аналоговый сигнал"
63 CRC16_modbus.c	Расчёт контрольной суммы CRC-16 Modbus
64 CRC16_modbus.h	Расчёт контрольной суммы CRC-16 Modbus
65 DDI.c	Опрос датчиков давления по Modbus
66 DDI.h	Опрос датчиков давления по Modbus
67 Modbus.c	Реализация протокола Modbus RTU

Имя файла	Назначение
68 Modbus.h	Реализация протокола Modbus RTU
69 ModbusLEDs.c	Модуль управления светодиодами индикаторами состояний "Отправка кадра" и "Приём кадра"
70 ModbusLEDs.h	Модуль управления светодиодами индикаторами состояний "Отправка кадра" и "Приём кадра"
71 ModbusListener.c	Модуль, перехватывающий сообщения Modbus (для полуконспекта, получающего данные в режиме прослушки)
72 ModbusListener.h	Модуль, перехватывающий сообщения Modbus (для полуконспекта, получающего данные в режиме прослушки)
73 ModbusTimer.c	Платформозависимая реализация таймеров Modbus
74 ModbusTimer.h	Платформозависимая реализация таймеров Modbus
75 ModbusUart.c	Платформозависимая реализация функций отправки/получения данных по UART для Modbus
76 ModbusUart.h	Платформозависимая реализация функций отправки/получения данных по UART для Modbus
77 DiscreteInputs.c	Модуль, опрашивающий дискретные входы
78 DiscreteInputs.h	Модуль, опрашивающий дискретные входы
79 DiscreteParameter.h	Модуль, опрашивающий дискретные входы
80 vds.c	Основной модуль, выполняющий функции ячейки ВДС
81 vds.h	Основной модуль, выполняющий функции ячейки ВДС
82 vds_SysDiagnostics.c	Модуль, формирующий сообщение SYS_DIAGNOSTICS ячейки ВДС
83 vds_SysDiagnostics.h	Модуль, формирующий сообщение SYS_DIAGNOSTICS ячейки ВДС

Имя файла	Назначение
84 vigilanceButtons.c	Модуль, обрабатывающий сигнал рукоятки бдительности машиниста
85 vigilanceButtons.h	Модуль, обрабатывающий сигнал рукоятки бдительности машиниста
86 diConfigTable.h	Описание параметров модуля дискретного ввода, определяемых конфигурационным числом от МПХ
87 VdsSignalsTable.h	Настройки входных сигналов ВДС
88 vdsTimings.h	Настройки временных интервалов для алгоритмов ВДС
89 Bootloader.c	API CAN-загрузчика для STM32F405
90 CAN_HAL.c	Платформозависимая реализация драйвера CAN
91 ChannelNumber.c	Платформозависимая реализация функции получения номера полукompлекта
92 DiscreteInputs_HAL.c	Платформозависимая реализация функций опроса состояния дискретных входов
93 DiscreteOutputs_HAL.c	Платформозависимая реализация функции управления дискретными выходами
94 MvdLED_HAL.c	Платформозависимая реализация функции управления светодиодной индикацией
95 SystemReset.c	Платформозависимая реализация функции сброса процессора
96 VirtualTimer_HAL.c	Платформозависимая реализация функций программных таймеров
97 Watchdog.c	Платформозависимая реализация функций сторожевого таймера
98 Bootloader.h	Платформозависимая реализация загрузчика
99 CAN_HAL.h	Платформозависимая реализация драйвера CAN
100 ChannelNumber.h	Платформозависимая реализация функции получения номера полукompлекта

Имя файла	Назначение
101 DiscreteInputs_HAL.h	Платформозависимая реализация функций опроса состояния дискретных входов
102 DiscreteOutputs_HAL.h	Платформозависимая реализация функции управления дискретными выходами
103 MvdLed_HAL.h	Платформозависимая реализация функции управления светодиодной индикацией
104 pin_out.h	Привязка дискретных сигналов к линиям GPIO микроконтроллера STM32F405
105 systemReset.h	Платформозависимая реализация функции сброса процессора
106 VirtualTimer_HAL.h	Платформозависимая реализация функций программных таймеров
107 Watchdog.h	Платформозависимая реализация функций сторожевого таймера

4 ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Блок ВВД (аппаратное средство) работает в составе ПО БЛОК-М и иных локомотивных систем.

Используемый микроконтроллер - STM32F205VC (ЦП – 120 МГц, ОЗУ – 96 Кб, ПЗУ – 256 Кб).

5 ВЫЗОВ И ЗАГРУЗКА

Программа начинает автоматически выполняться при подаче электропитания на блок ВВД.

6 ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

Входными данными для ПО ВВД являются:

- сообщения MCO_STATE_A, MCO_LIMITS_A от ячейки ЦО – для первого полукомплекта;
- сообщения MCO_STATE_B, MCO_LIMITS_B от ячейки ЦО – для второго полукомплекта;
- сообщения SAUT_TORM_A, SAUT_UST_DD_A от модуля BC-CAУТ – для первого полукомплекта;
- сообщения SAUT_TORM_B, SAUT_UST_DD_B от модуля BC-CAУТ – для второго полукомплекта;
- сообщения SYS_DIAGNOSTICS от сервисного оборудования;
- сообщения от датчиков давления;
- состояния дискретных входов.

Выходными данными для ПО ВВД являются:

- состояние выходных ключей (коммутируемых цепей);
- сообщения MVD_STATE_A, VDS_STATE_A, GOT_A, MVD_DD, AUX_RESOURCE_MVD_A, AUX_RESOURCE_VDS_A – для первого полукомплекта;
- сообщения MVD_STATE_B, VDS_STATE_B, GOT_B, AUX_RESOURCE_MVD_B, AUX_RESOURCE_VDS_B – для второго полукомплекта.

ПРИЛОЖЕНИЕ 1
(Справочное)

ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ

Блок ВВД – блок ввода и вывода дискретных сигналов;

БЛОК-М – безопасный локомотивный объединенный комплекс масштабируемый;

КОН – блок контроля несанкционированного отключения ЭПК ключом;

ПО – программное обеспечение;

САУТ – система автоматического управления торможения поезда;

ЦО – модуль центрального обработчика;

ЭПК – электропневматический клапан дистанционного управления торможением.

ПРИЛОЖЕНИЕ 2

(Обязательное)

СБОРКА ПРОГРАММНОГО МОДУЛЯ ВВОДА И ВЫВОДА ДИСКРЕТНЫХ СИГНАЛОВ

1 Цель сборки

В результате сборки будет получен:

- исполняемый файл программного модуля ВВД версии «3» подверсии «7» с наименованием «ВВД ver.3.7.sfp», контрольной суммой CRC32 равной «DC9309F4», предназначенный для исполнения на микроконтроллере «STM32F205VC»;
- исполняемый файл программного модуля ВВД версии «3» подверсии «10» с наименованием «ВВД ver.3.10.sfp», контрольной суммой CRC32 равной «06D26E00», предназначенный для исполнения на микроконтроллере «STM32F205VC».

2 Необходимые технические средства

Для сборки исполняемого файла программного модуля центрального вычислителя необходимы технические средства со следующими минимальными характеристиками: ПЭВМ с частотой центрального процессора 1,5 ГГц, объемом оперативной памяти 1 Гбайт, объемом жесткого диска 50 Гбайт.

3 Необходимые программные средства

Сборка исполняемого файла программного модуля ВВД выполняется с помощью среды разработки «IAR Embedded Workbench for ARM» версии 7.50.2.10505, формирующей бинарный исполняемый файл, а также программы FmPack версии 2.12 (или выше), упаковывающей данный исполняемый файл в архив формата «.sfp».

4 Подготовка к сборке

- Скопировать папку «armVvdFlexible» с проектом из места хранения;
- Запустить среду разработки «IAR Embedded Workbench for ARM», затем в основном меню выбрать File -> Open -> Workspace. В открывшемся окне выбрать файл «Project.eww» (рисунок 2.1).

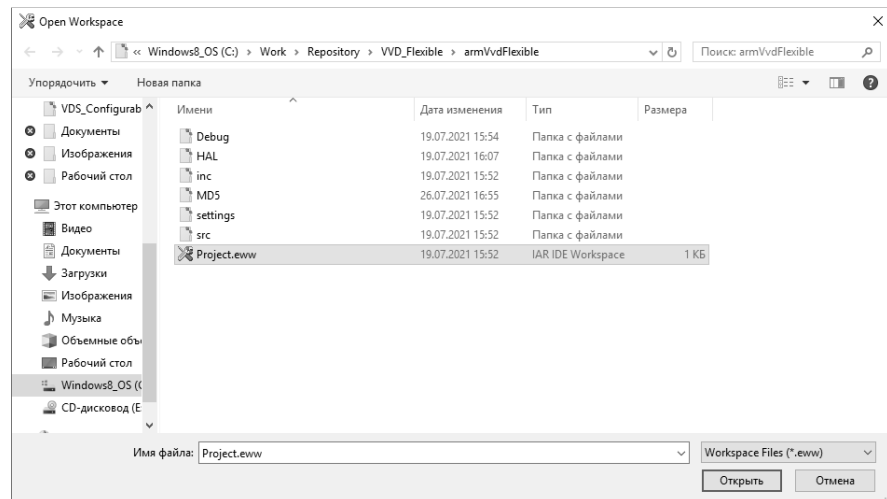


Рисунок 2.1 - Окно выбора файла «Project.eww»

5 Выполнение сборки

Дождаться открытия проекта (рисунок 2.2).

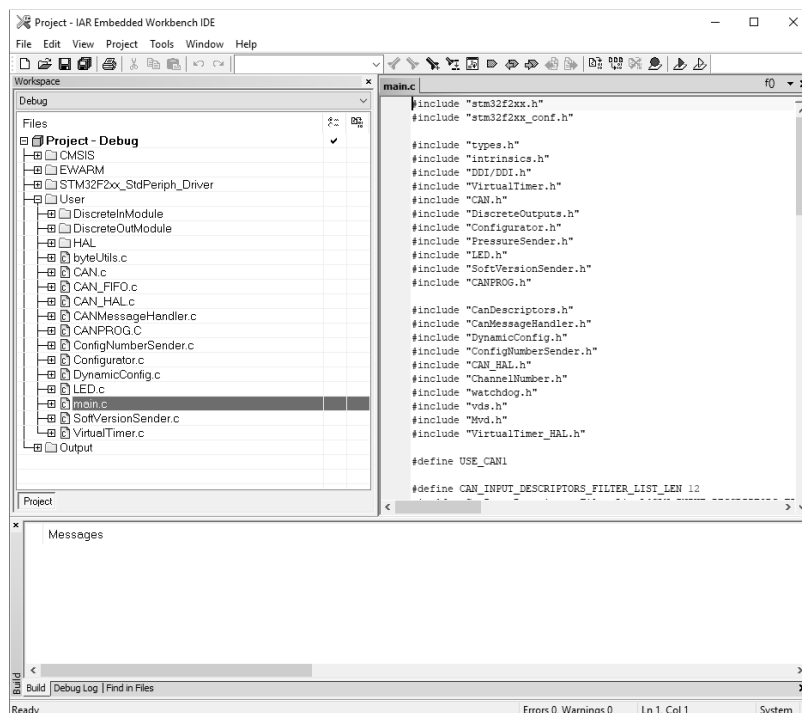


Рисунок 2.2 - Окно среды разработки с открытым проектом

Далее:

- Запустить сборку, выбрав на главном меню команду Project -> Rebuild All;
- Дождаться завершения сборки (сообщения среды разработки о завершении сборки без ошибок «Total number of errors: 0») (рисунок 2.3);

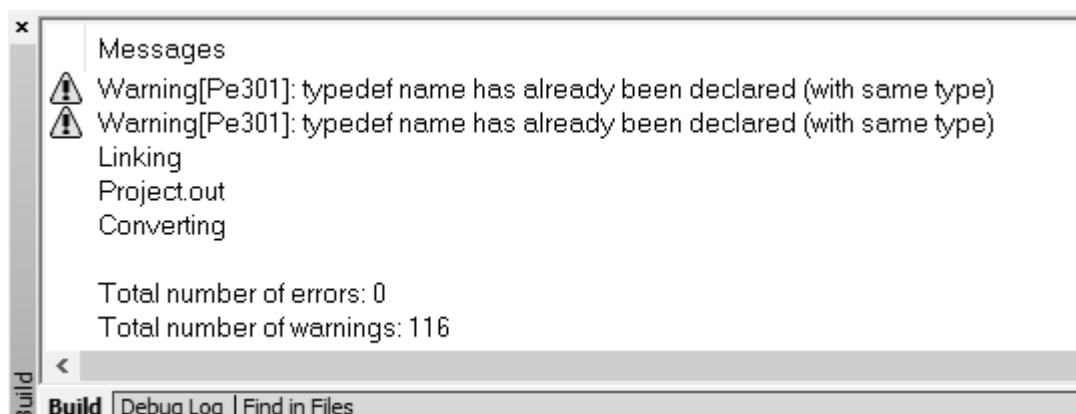


Рисунок 2.3 - Сообщения среды разработки

- Упаковать бинарный файл в архив .sfr.

Для этого запустить программу «FmPack.exe», в открывшемся окне нажать кнопку «Открыть файл проекта» и выбрать файл «vvd.fpc», расположенный в папке проекта «armVvdFlexible» (рисунок 2.4);

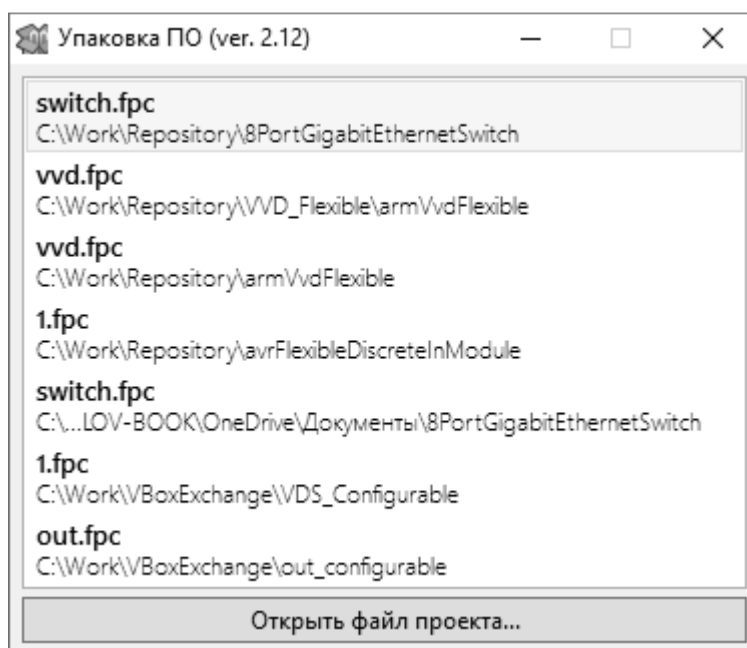


Рисунок 2.4 - Окно выбора файла проекта

- В открывшемся окне в поле «Версия» вписать требуемый номер версии. Нажать кнопку «Сохранить» (рисунок 2.5 – номер версии указан для примера);

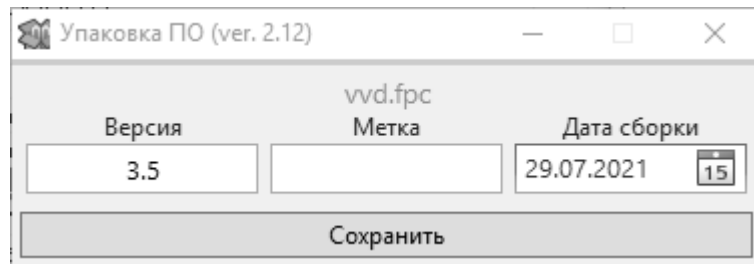


Рисунок 2.5 - Окно упаковки ПО

- В открывшемся меню выбрать папку, в которую требуется поместить результирующий файл архива «.sfp» и нажать кнопку «Сохранить» (рисунок 2.6).

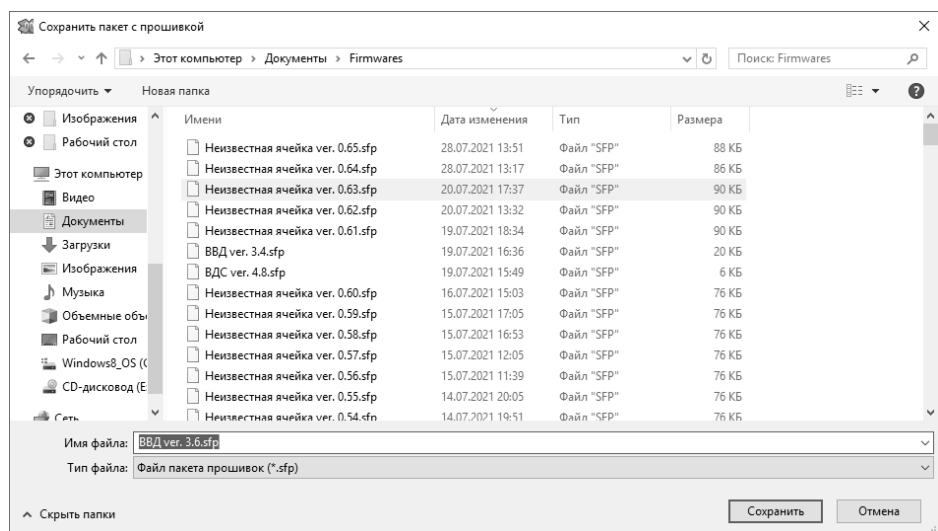


Рисунок 2.6 - Окно сохранения файла формата «.sfp»

[illegible]