

СидПроф

командный программатор

Краткое описание

Содержание

1. Введение	3
2. Работа со списками проектов	5
3. Работа с проектами	6
4. Работа с конфигурациями	8
5. Работа с командами	9
6. Программирование устройства	14
7. Локализация	18
8. Настройки программы	19
9. Аппаратная часть (базовая версия)	21

1. Введение

1.1. Программатор CmdProg предназначен для программирования микроконтроллеров, микросхем памяти, а также других микросхем с интерфейсами I2C, SPI и т.д.

В описании программного обеспечения программируемая микросхема называется «устройство».

1.2. Основные характеристики программатора:

- командно-ориентированный интерфейс пользователя;
- связь программатора с компьютером через последовательный интерфейс (COM-порт);
- программирование нескольких устройств поочередно или одновременно;
- программирование устройства с учётом сигнатуры;
- установка выходов программатора после программирования в требуемое состояние;
- возможность добавления новых одноклассовых устройств;
- возможность создания базы проектов;
- возможность использования национального языка (локализация);
- поддерживаемые операционные системы: Windows (7 и выше), Linux.

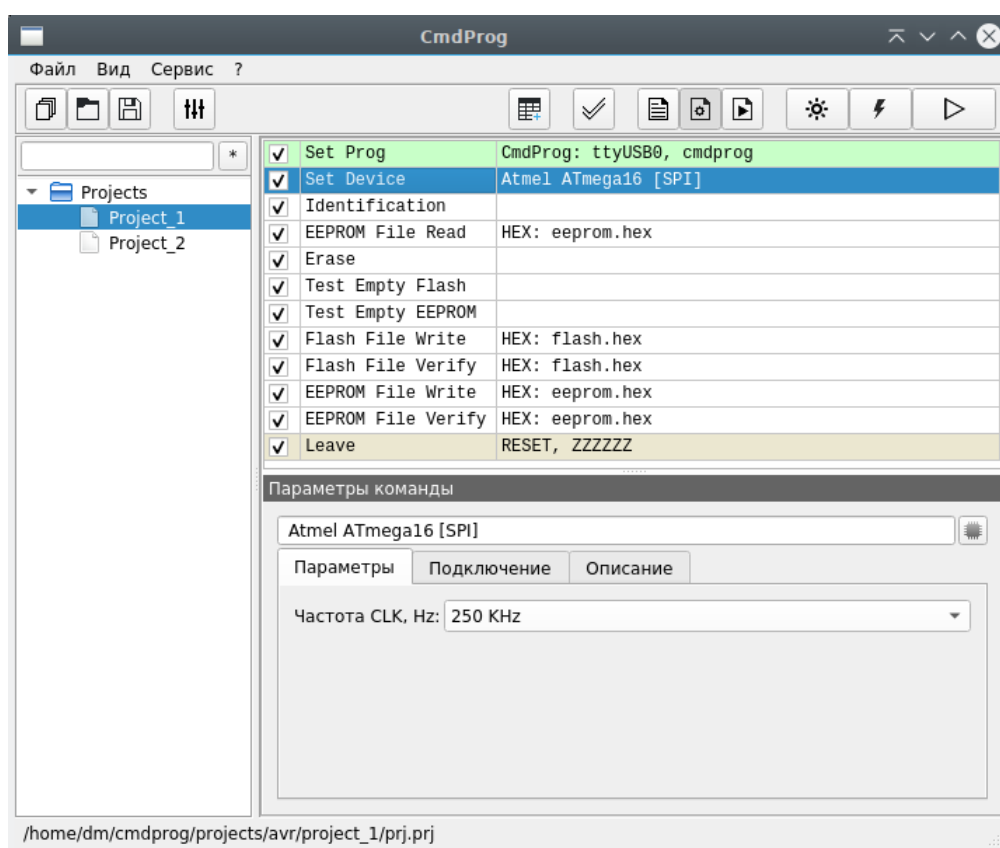


Рисунок 1.1. Главное окно программы

1.3. Программатор ориентирован на обновление программного обеспечения оборудования, где требуется последовательное выполнение нескольких команд, например:

1	Чтение EEPROM	сохранение настроек оборудования
2	Очистка памяти	
3	Проверка на пустоту	
4	Запись Flash	запись новой версии программного обеспечения
5	Верификация Flash	
6	Запись EEPROM	восстановление настроек оборудования
7	Верификация EEPROM	
8	Завершение программирования	установка выходов программатора в требуемое состояние

1.4. Описание элементов управления главного окна программы.

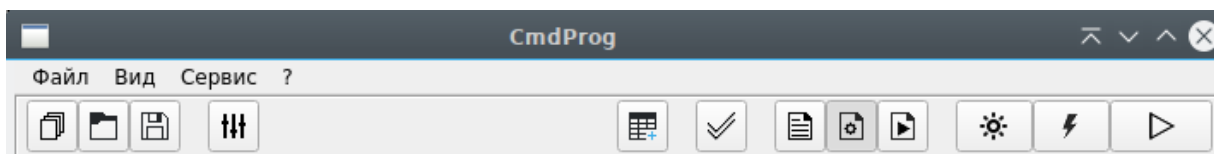














Рисунок 1.2. Элементы управления главного окна программы

-  - открыть окно выбора списка проектов
-  - открыть список проектов
-  - сохранить проект
-  - открыть окно «Настройки программы»
-  - открыть/закрыть окно «Селектор команд»
-  - скрыть/показать невыбранные команды
-  - показать описание проекта
-  - показать параметры команды
-  - показать ход выполнения
-  - визуальная идентификация программатора
-  - сформировать сигнал «RESET» для устройства
-  - выполнить конфигурацию

1.5. Структура файлов программы.

- | | |
|-------------|--|
| \devices | - директория с файлами описания программируемых устройств |
| \language | - директория с файлами локализации |
| \progs | - директория с файлами конфигурации программаторов |
| \projects | - директория с файлами проектов |
| cmdprog | - выполняемый файл программы |
| cmdprog.hk | - конфигурационный файл с описанием горячих клавиш |
| cmdprog.ini | - конфигурационный файл настроек программы |
| cmdprog.lpl | - конфигурационный файл списков проектов |
| crc.lst | - файл-список с перечнем алгоритмов расчёта контрольных сумм |

2. Работа со списками проектов

2.1. Списки проектов служат для группировки проектов по смысловому значению. Директория, в которой находится файл списка проектов (*.lst) является текущей для проектов при задании пути. Управление списками проектов осуществляется через контекстное меню в окне «Выбор списка проектов», либо через меню «Файл» - «Список проектов».

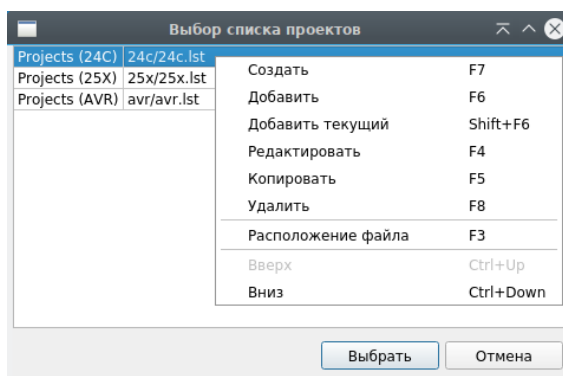


Рисунок 2.1. Окно «Выбор списка проектов»

Описание пунктов контекстного меню выбора списка проектов:

Создать	- создать новый список проектов
Добавить	- добавить существующий список проектов
Добавить текущий	- добавить в список проектов открытый на данный момент список
Редактировать	- редактировать список проектов
Копировать	- копировать список проектов
Удалить	- удалить список проектов
Расположение файла	- открыть директорию с файлом списка проектов
Вверх	- переместить строку по списку вверх
Вниз	- переместить строку по списку вниз

2.2. Название списка проектов может быть отображено в главном окне программы: «Сервис» - «Настройки» - «Интерфейс» - «Отображать наименование списка проектов».

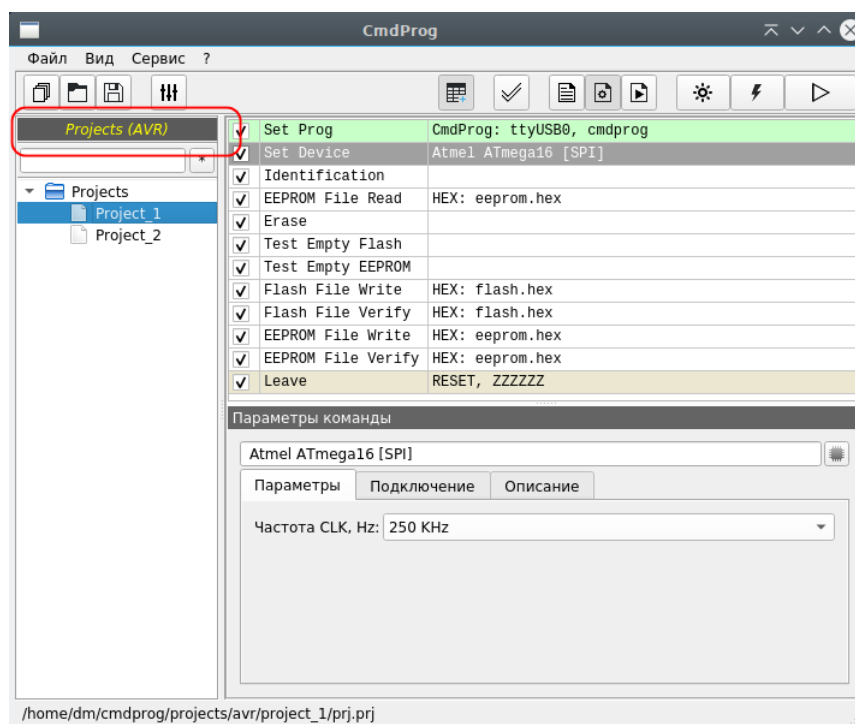


Рисунок 2.2. Отображение названия списка проектов

3. Работа с проектами

3.1. Проекты всегда принадлежат какому-либо списку проектов. Управление проектами осуществляется через контекстное меню в списке проектов, либо через меню «Файл» - «Проект». Каждый проект рекомендуется располагать в отдельной директории.

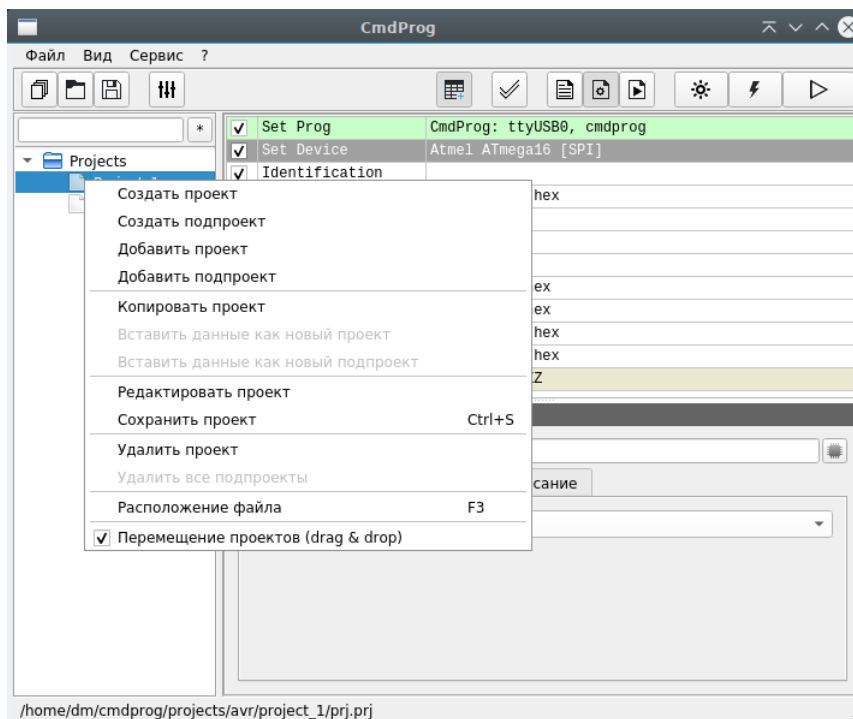


Рисунок 3.1. Контекстное меню списка проектов

3.2. Список проектов может быть дополнен фильтром. Фильтрация проектов осуществляется по заданной маске. Кнопка с символом «*» справа от поля ввода сбрасывает фильтр. Отображение фильтра проектов в главном окне программы производится через настройки программы: «Сервис» - «Настройки» - «Интерфейс» - «Отображать строку фильтра».

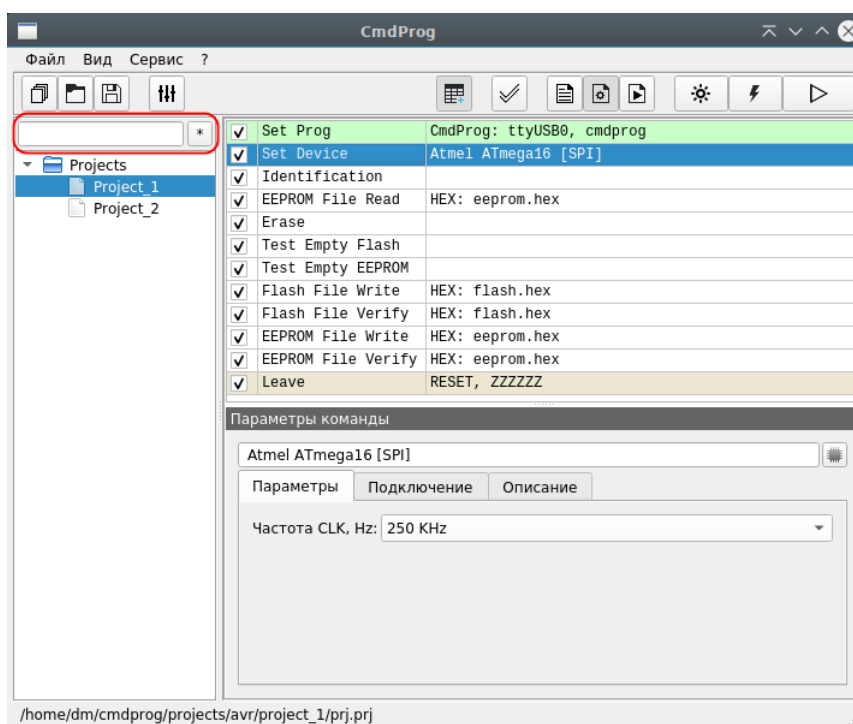


Рисунок 3.2. Фильтр проектов

3.3. Копирование проектов работает между списками проектов, т.е. можно скопировать проект в одном списке проектов и вставить в другой список проектов.

3.4. Файл проекта располагается в директории проекта и имеет расширение «*.prj». Файлы, относящиеся к данному проекту, по умолчанию, также располагаются в директории проекта.

4. Работа с конфигурациями

4.1. Конфигурации группируют команды для последующего выполнения. Каждый проект содержит минимум одну конфигурацию. Количество конфигураций в проекте неограниченно.

Использование конфигурации позволяет:

- программировать нескольких устройств поочерёдно (команда «JAMP»);
- программировать несколько устройств одновременно (команда «EXECUTE»);
- программировать устройство с учётом сигнатуры (команда «Auto Detect»).

4.2. Управление конфигурациями осуществляется через контекстное меню в списке команд, либо через меню «Файл» - «Конфигурация». Вызов контекстного меню для работы с конфигурациями производится нажатием правой кнопки мыши в списке команд вместе с клавишей «CTRL».

4.3. Порядок конфигураций может быть изменён перемещением вкладок.

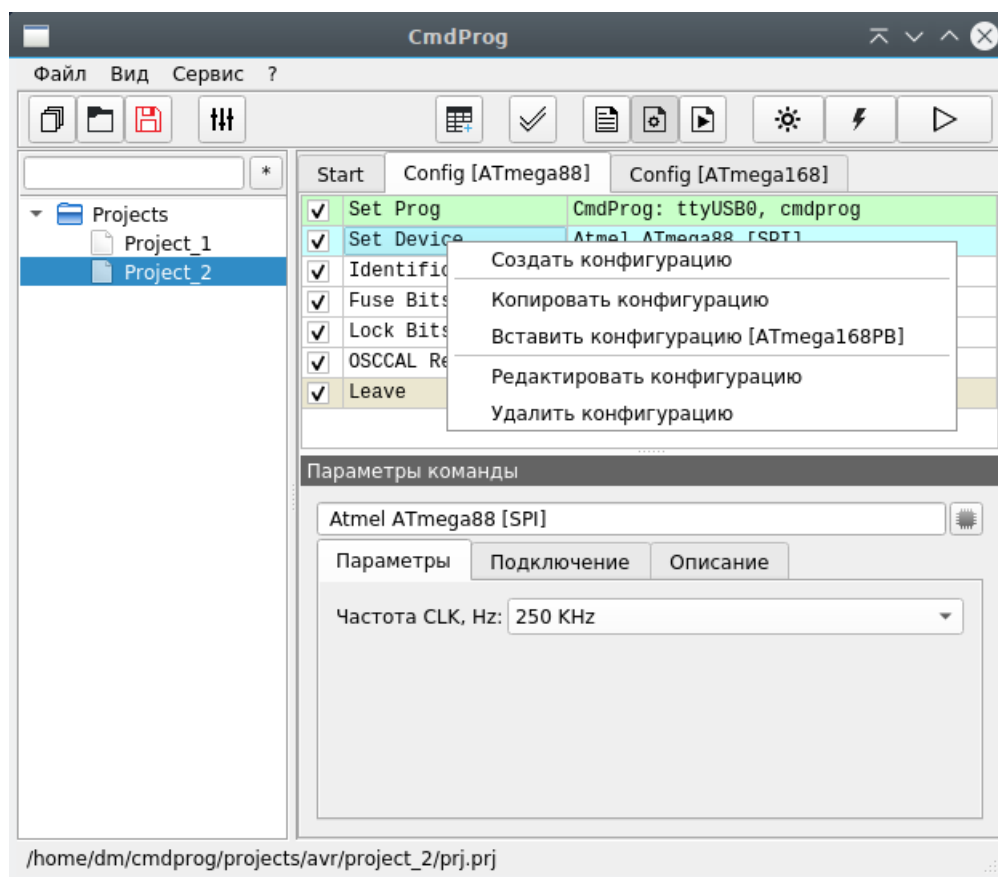


Рисунок 4.1. Проект с конфигурациями

5. Работа с командами

5.1. Каждое выполняемое действие задаётся с помощью отдельной команды. Команда всегда принадлежит одной из конфигурации. Добавление команд производится с помощью «Селектора команд». Все команды делятся на три группы: «Tools», «Prog», «Device». Команды в «Селекторе команд» также разделены на три группы. Вызов «Селектора команд» производится через меню «Файл» - «Команда» - «Селектор команд», либо нажатием кнопки в главном окне программы.

5.2. Управление командами в списке команд производится через контекстное меню в панели команд, либо через меню «Файл» - «Команда».

5.3. Для улучшения восприятия списка команд часть команд групп «Tools» и «Prog» имеют цветовую окраску. Цветовую окраску можно отключить или изменить через настройки программы: «Сервис» - «Настройки» - «Интерфейс».

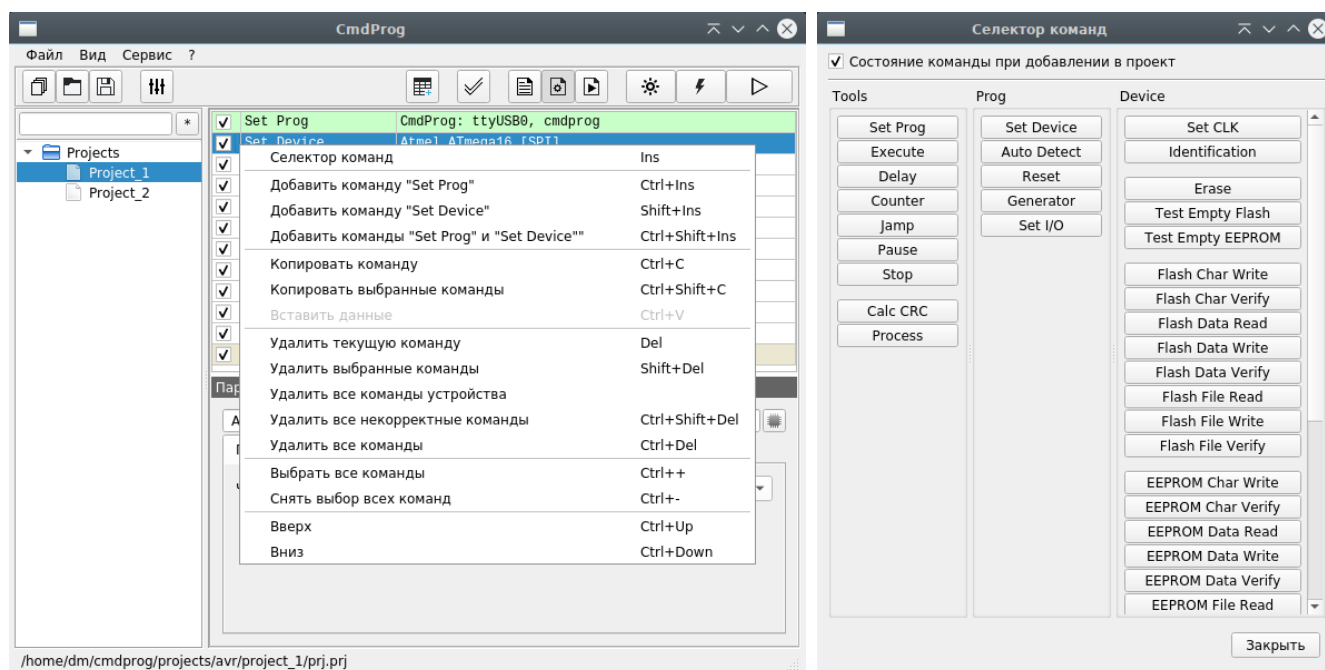


Рисунок 5.1. Контекстное меню панели команд и окно «Селектор команд»

5.4. **Команды группы «Tools»** не взаимодействуют с программатором (кроме команды «Set Prog») и программируемым устройством. Данные команды выполняют сервисные функции. Группа «Tools» включает в себя следующие команды:

Команда	Описание
Set Prog	установка типа программатора и параметров подключения
Execute	запуск конфигурации на выполнение в фоновом режиме
Delay	задержка
Counter	счётчик
Jump	переход на определённую команду в конфигурации
Pause	приостановка выполнения конфигурации с возможностью продолжения или останова
Stop	останов выполнения конфигурации
Calc CRC	расчёт контрольной суммы файла по заданному алгоритму
Process	запуск внешней программы

5.5. Команды группы «Prog» предназначены для управления программатором, и не взаимодействуют с программируемым устройством (кроме команд «Set Device» и «Auto Detect»). Список поддерживаемых программатором команд группы «Prog» задаётся в файле конфигурации программатора «progs/*.cfg». Полный перечень команд группы «Prog» выглядит следующим образом:

Команда	Описание
Set Device	установка типа программируемого устройства и параметров программирования
Auto Detect	определение сигнатуры устройства и переход на выполнение соответствующей конфигурации
Reset	формирование импульса сброса с заданными параметрами (полярность, длительность)
Generator	формирование внешней частоты с заданными параметрами (частота, скважность)
Set I/O	установка выходов программатора в требуемое состояние
Set Vcc	установка напряжения питания устройства
Set Vpp	установка высокого напряжения программирования
Set Vicc	установка напряжения и тока питания устройства
Set Vlpp	установка высокого напряжения и тока программирования

5.6. Команды группы «Device» предназначены для непосредственного взаимодействия с программируемым устройством. Перечень команд группы «Device» зависит от конкретного устройства.

Пример. Перечень команд устройств серии 24Схх (память I2C):

Команда	Описание
Set CLK	установка скорости передачи по интерфейсу I2C
Set Address	установка адреса устройства на шине I2C
Erase	очистка устройства
Test Empty	тестирование на пустоту
Char Write	запись константы
Char Verify	верификация с константой
Data Read	чтение блока данных (несколько байт)
Data Write	запись блока данных (несколько байт)
Data Verify	верификация блока данных (несколько байт)
File Read	чтение данных из устройства в файл
File Write	запись данных в устройство из файла
File Verify	верификация данных устройства с данными в файле
Leave	выход из режима программирования, установка выходов программатора в требуемое состояние

5.7. Первой командой в конфигурации должна быть любая команда из группы «Tools». Как правило, это команда «Set Prog». Данная команда определяет параметры программатора. Без данной команды программирование невозможно. Команды группы «Tools» могут располагаться в любом месте конфигурации.

5.8. Второй командой в конфигурации может быть любая команда из группы «Prog». Для установки типа устройства предназначена команда «Set Device», которая выполняет следующие действия:

- установка параметров программирования;
- инициализация интерфейса программирования;
- ввод устройства в режим программирования (зависит от типа устройства).

5.9. После команды «Set Device» могут быть добавлены команды из группы «Device».

5.10. Процесс программирования должен завершаться командой «Leave». В случае отсутствия команды «Leave» в списке, команда «Leave» будет выполнена принудительно с параметрами по умолчанию для данного устройства (как правило, с установкой выходов программатора в высокоимпедансное состояние).

5.11. Последовательность команд при программировании микросхемы памяти SPI (W25Q128JV) приведена на рисунке 5.2.

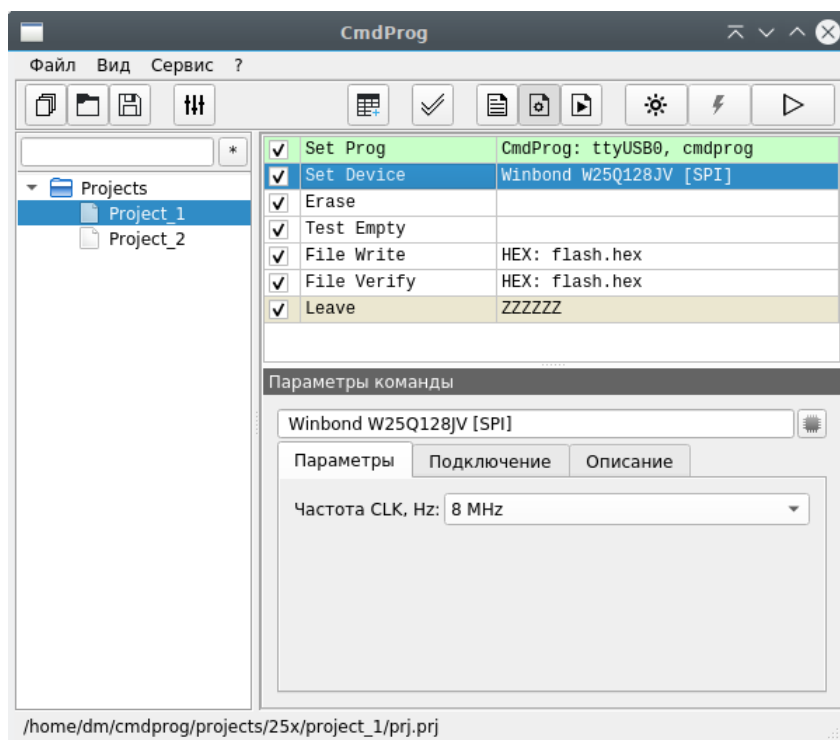


Рисунок 5.2. Главное окно программы

5.12. Допускается в одной конфигурации наличия нескольких команд «Set Prog» и «Set Device». Пример использования нескольких команд «Set Prog» и «Set Device» приведён на рисунке 5.3.

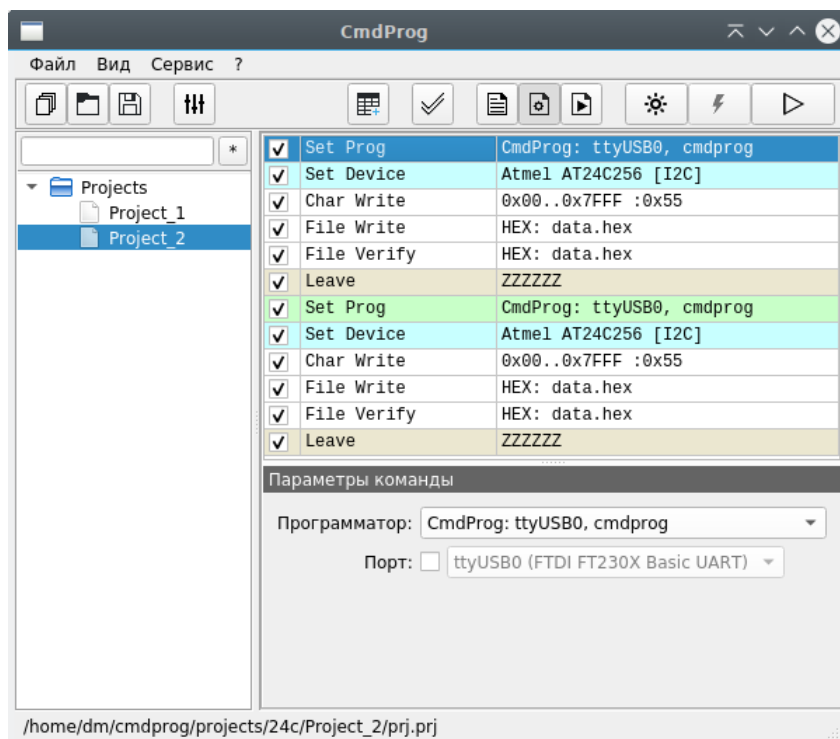


Рисунок 5.3. Конфигурация с несколькими командами «Set Prog» и «Set Device»

5.13. Команды групп «Prog» и «Device» отображаются в «Селекторе команд» с учетом текущей команды в списке команд. «Селектор команд» знает информацию только о текущей команде и о командах, расположенных выше текущей:

- если в списке команд отсутствуют команды (создана новая конфигурация), то в «Селекторе команд» будут отсутствовать команды групп «Prog» и «Device»;
- если текущая команда «Set Prog», то в «Селекторе команд» будут отсутствовать команды группы «Device», т.к. команда «Set Device» расположена ниже текущей команды;
- если команда некорректная, то в «Селекторе команд» будут отсутствовать команды групп «Prog» и «Device».

5.14. Команды могут быть некорректными, т.е. для них будет отсутствовать информация о программаторе или устройстве, либо команды не поддерживаются программатором или устройством. Например, если команда «Set Device» находится в списке без команды «Set Prog», то команда «Set Device» и следующие за ней команды будут некорректными, т.к. отсутствуют данные о программаторе. Некорректные команды отображаются красным цветом с зачёркиванием (см. рисунок 5.4).

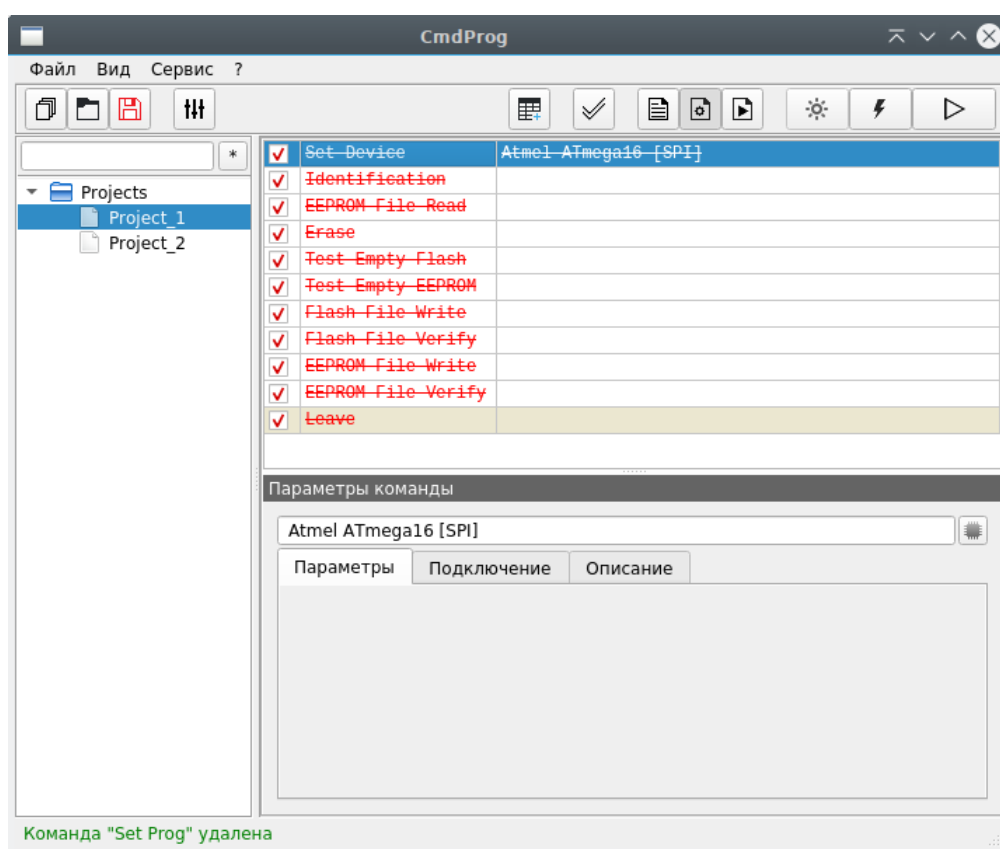


Рисунок 5.4. Отображение некорректных команд

5.15. Если команда является корректной, но имеет ошибки в параметрах команды, то такая команда отображается в списке команд красным цветом. Поля ввода с неправильными значениями параметров, также окрашены в красный цвет. Пример команды с ошибками в параметрах приведён на рисунках 5.5 и 5.6.

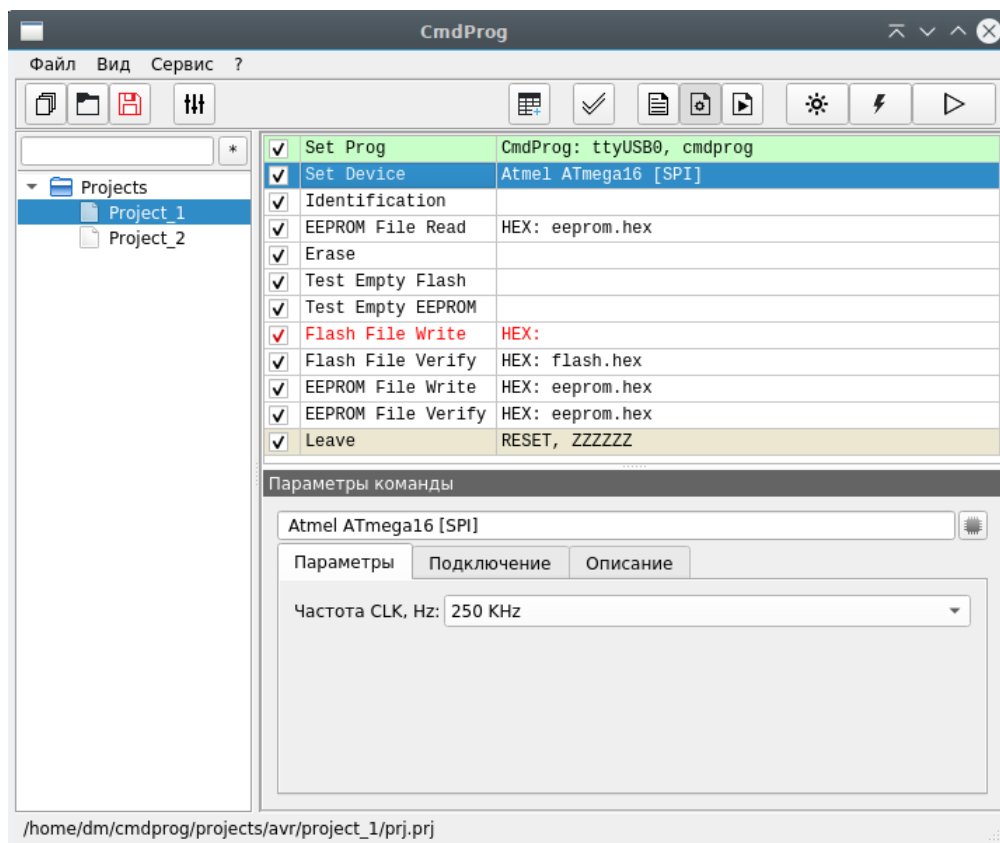


Рисунок 5.5. Отображение команды с ошибками в параметрах

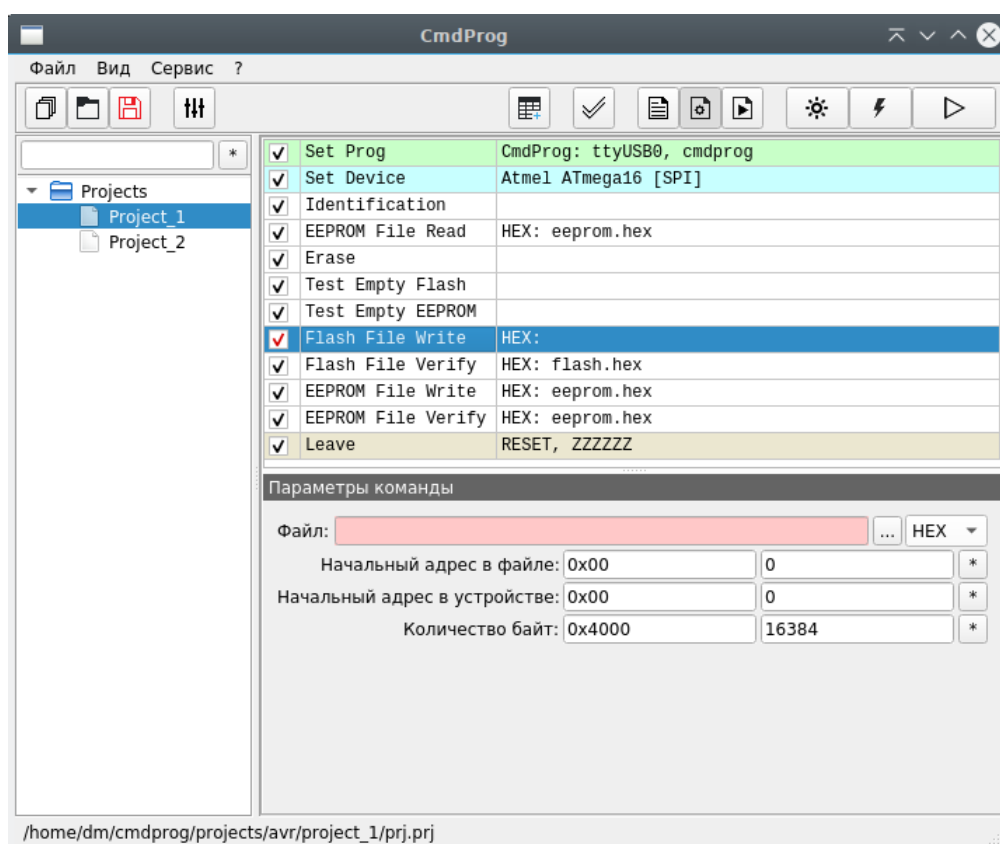


Рисунок 5.6. Отображение команды с ошибками в параметрах

6. Программирование устройства

6.1. Процесс программирования представляет собой последовательное выполнение команд *текущей* конфигурации. Выполняться будут команды, напротив которых установлен признак выбора «☒».

6.2. Навигация по элементам управления, в рамках проекта, не прерывает ход выполнения (программирования).

6.3. Программирование (общая часть).

6.3.1. Выполнение конфигурации производится нажатием на кнопку «▶». При этом кнопка меняет изображение на «■» (см. рисунок 6.1). При нажатии на кнопку «■» выполнение конфигурации прерывается.

После запуска конфигурации на выполнение, в зависимости от настроек программы, будет открыта вкладка «Выполнение». Переключение между вкладками «Описание проекта», «Параметры команды», «Выполнение» производится с помощью соответствующих кнопок (отмечены синим цветом на рисунке 6.1).

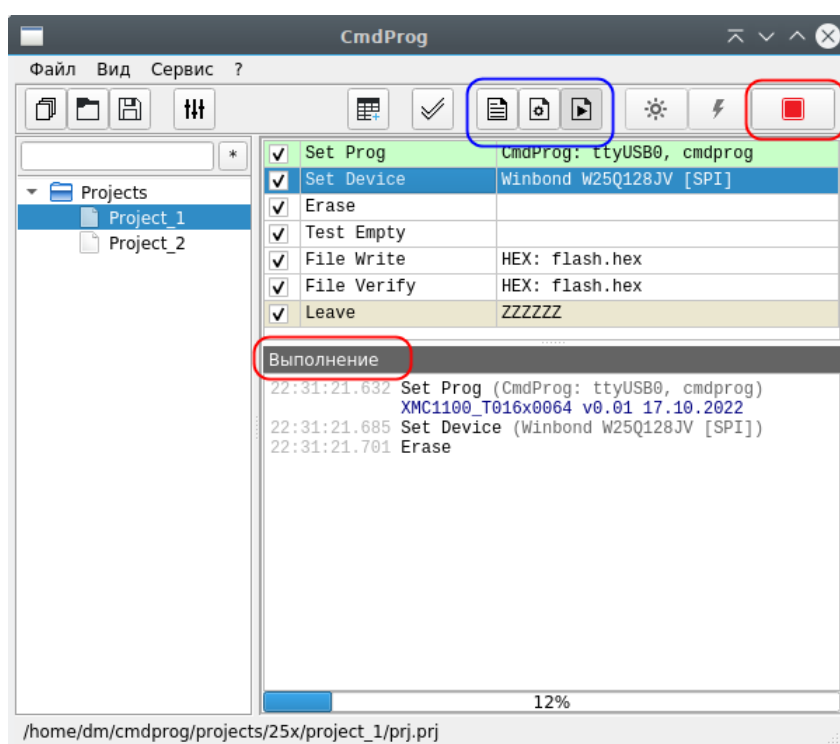



Рисунок 6.1. Ход выполнения

6.3.2. Если в списке команд присутствует команда «Pause», то выполнение данной команды приводит к приостановке выполнения конфигурации. После выполнения команды «Pause» изображение кнопки выполнения меняется на «▶». Повторное нажатие на кнопку выполнения продолжит выполнение конфигурации. Если требуется завершить выполнение конфигурации, то необходимо нажать на кнопку выполнения вместе с клавишей CTRL.

6.3.3. Если устройство поддерживает сигнал «RESET», то можно сформировать данный сигнал, нажав на кнопку «⚡». При нажатии на данную кнопку в программатор будет отправлена команда «RESET» с параметрами для данного устройства (полярность, длительность). При этом в списке команд должна быть текущей команда, относящаяся к программируемому устройству (команда «Set Device» и ниже). При выполнении команды «RESET», в зависимости от настроек программы, будет открыта вкладка «Выполнение».

Если устройство не поддерживает сигнал «RESET», то кнопка «⚡» будет неактивна.

6.3.4. Если используется несколько программаторов подключенных на разные последовательные порты, то для определения принадлежности программатора к текущей конфигурации можно нажать на кнопку «» (визуальная идентификация программатора). При нажатии на данную кнопку в программатор отправляется команда «ID Prog». Выполнение команды приводит к свечению/миганию индикаторов на плате программатора. Комбинация свечения/мигания индикаторов определяется разработчиком программатора. При этом в списке команд должна быть выбрана команда, относящаяся к программатору (команда «Set Prog» и ниже). При выполнении команды «ID Prog», в зависимости от настроек программы, будет открыта вкладка «Выполнение».

6.3.5. Программное обеспечение позволяет работать с несколькими программаторами поочерёдно или одновременно, используя конфигурации. Каждому программатору назначается свой последовательный порт (СОМ-порт). При выполнении программирования, индикация на плате программатора будет дублироваться подсвечиванием имени конфигурации соответствующим цветом (см. рисунок 6.2):

- зелёный – конфигурация выполнена успешно;
- синий – идёт выполнение конфигурации;
- красный – конфигурация выполнена неудачно.

Перед повторным выполнением конфигураций подсвечивание имён конфигураций может быть сброшено: «Файл» - «Выполнение» - «Сбросить статус выполнения всех конфигураций».

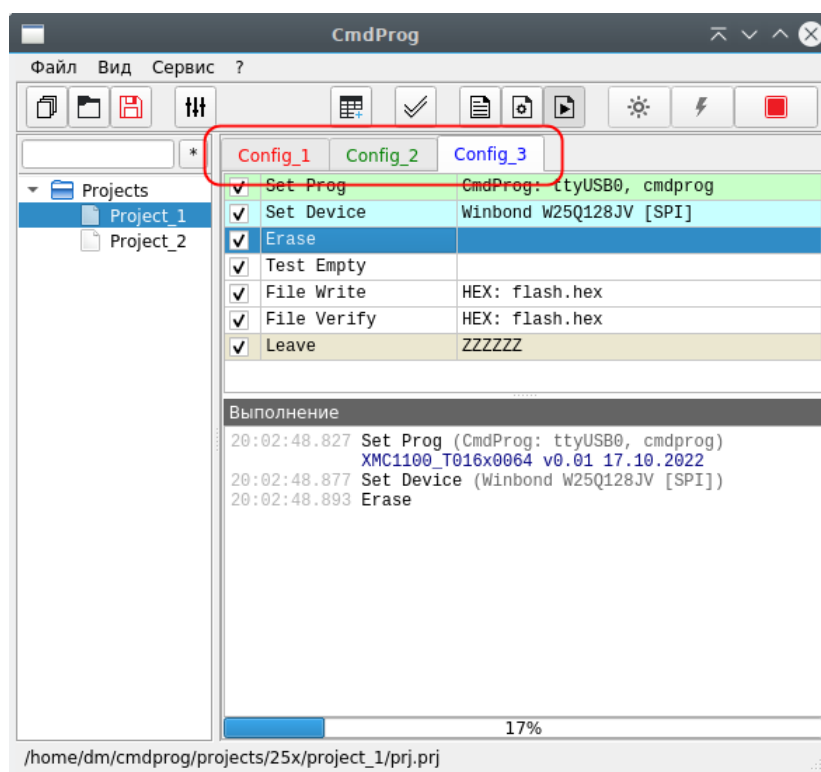


Рисунок 6.2. Главное окно программы

6.4. Поочерёдное программирование устройств (команда «Jatr»).

6.4.1. Рассмотрим пример программирования восьми штук микросхем памяти 24C02 подключенных к шине I2C и имеющих разные адреса (A2, A1, A0). Для этого создадим восемь вкладок. Список команд в каждой конфигурации одинаковый, отличаются только параметры команд «Set Device» (адрес устройства) и «Jump» (указатель на конфигурацию). После выполнения конфигурации с именем «0» по команде «Jump» передача выполнения перейдёт конфигурации с именем «1» и т.д. После выполнения конфигурации с именем «7» процесс выполнения закончится. Если открыть конфигурацию с именем «6» и нажать кнопку выполнения, то будут выполнены поочерёдно конфигурации «6» и «7». Главное окно программы для данного примера приведено на рисунке 6.3.

6.4.2. Ход выполнения будет отображаться во вкладке «Выполнение» конфигурации, которая была текущей на момент запуска выполнения.

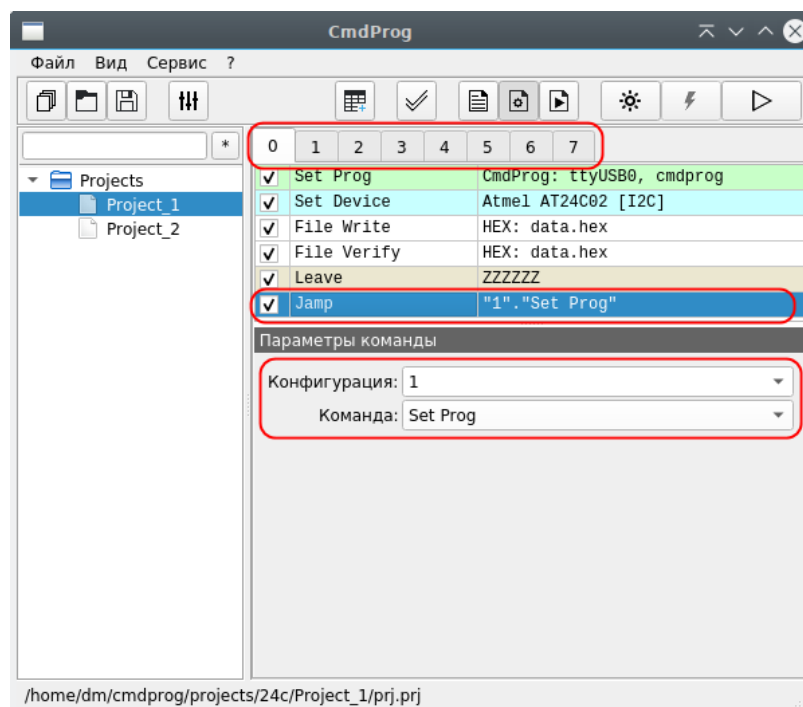


Рисунок 6.3. Поочерёдное программирование устройств

6.5. Одновременное программирование устройств (команда «Execute»).

6.5.1. Для одновременного программирования нескольких устройств необходимо:

- создать конфигурации для каждого программируемого устройства;
- для каждой команды «Set Prog» указать соответствующий последовательный порт;
- создать конфигурацию и добавить в неё требуемое количество команд «Execute»;
- в параметрах команд «Execute» указать конфигурацию для выполнения;
- запустить на выполнение конфигурацию, содержащую команды «Execute».

6.5.2. Конфигурации, запускаемые командой «Execute», будут выполняться в фоновом режиме одновременно. Пример конфигурации приведён на рисунке 6.4.

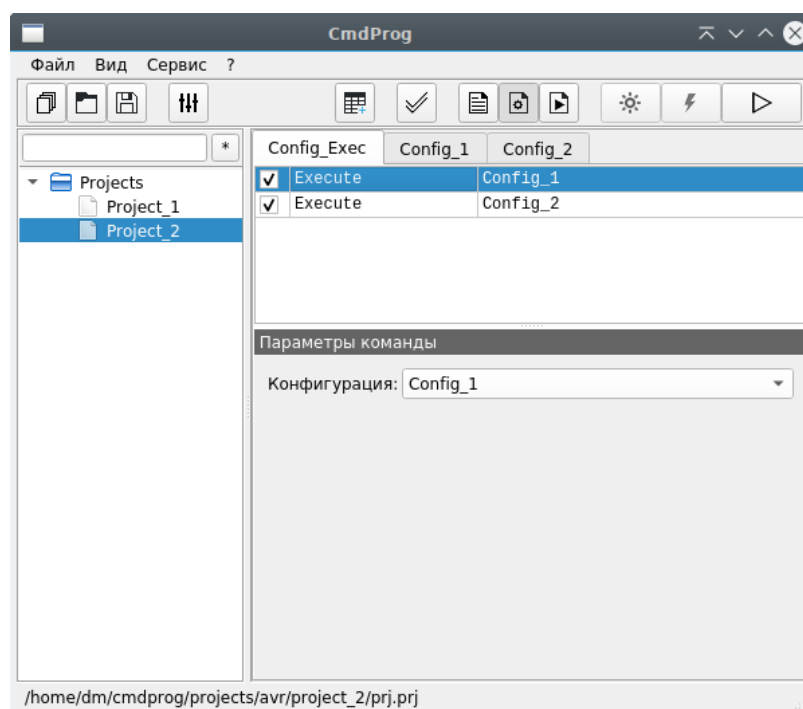


Рисунок 6.4. Одновременное программирование нескольких устройств

6.5.3. Ход выполнения каждой конфигурации будет отображаться в своей вкладке «Выполнение».

6.5.4. Если в процессе выполнения конфигураций производится чтение данных из устройства в файл, то необходимо исключить запись в один и тот же файл разными конфигурациями. Для исключения конфликта можно указать разные имена файлов в параметрах команд, либо хранить файлы каждой конфигурации в своей директории.

6.5.5. В случае если необходимо запрограммировать только одно устройство, можно перейти в соответствующую конфигурацию и нажать кнопку выполнения. Дополнительная корректировка проекта при этом не требуется. Также можно выбрать какие из команд «Execute» должны выполняться, установив или сняв признак выбора команды «☒».

6.6. Программирование с учётом сигнатуры устройства (команда «Auto Detect»).

6.6.1. Программирование в данном режиме подразумевает под собой определение сигнатуры устройства и передачу выполнения соответствующей конфигурации в зависимости от сигнатуры. Для использования данного режима необходимо, чтобы все устройства относились к одной группе (способ определения сигнатуры).

Для программирования устройства с учётом сигнатуры необходимо:

- создать конфигурации для каждого программируемого устройства;
- создать конфигурацию и добавить в неё команды «Set Prog» и «Auto Detect»;
- в параметрах команды «Auto Detect» выбрать группу устройств, указать соответствие устройств (сигнатуры) и конфигураций, определить параметры «Jump» и «Skip»;
- запустить на выполнение конфигурацию, содержащую команду «Auto Detect».

6.6.2. Пример конфигурации для программирования устройства с учётом сигнатуры приведён на рисунке 6.5.

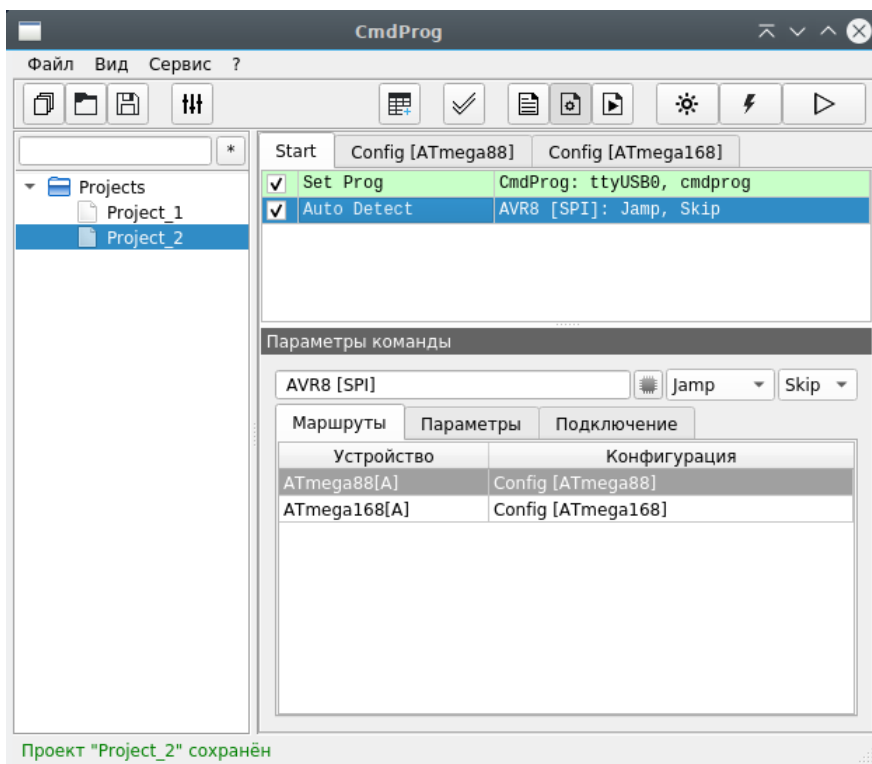


Рисунок 6.5. Программирование устройства с учётом сигнатуры

7. Локализация.

7.1. Программное обеспечение позволяет перевести интерфейс программы на национальный язык. Файлы, содержащие фразы на национальном языке, размещаются в директории «language». Фразы в файле позиционно-зависимые. Закомментированные строки начинаются с символа «#».

7.2. Описание позиций в файле (*.lng):

- 1 строка – наименование языка на английском языке;
- 2 строка – наименование языка на национальном языке;
- 3 строка ... – фразы на национальном языке.

7.3. Интерфейс окна выбора языка отображается только на английском языке.

7.4. После смены языка требуется перезапуск программы.

7.5. В файлах локализации (*.lng) используется кодировка UTF-8 (65001).

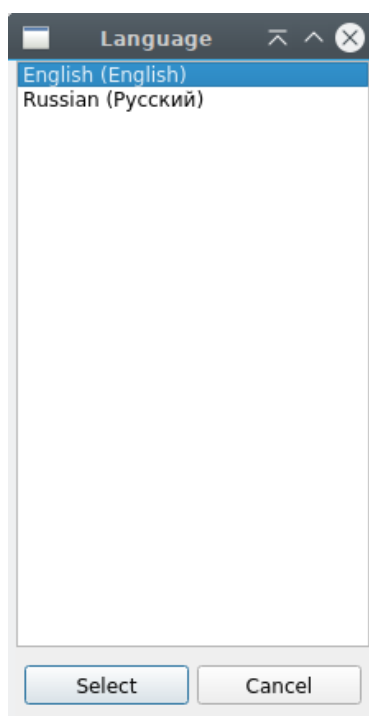


Рисунок 7.1. Окно выбора языка интерфейса программы

8. Настройки программы

8.1. Настройки программы разделены на четыре группы:

- Основные - основные (общие) параметры программы
- Программаторы - настройка типов и псевдонимов программаторов
- Интерфейс - параметры отображения интерфейса программы
- Подтверждения - установка подтверждений при выполнении действий

8.2. **Настройки программы. Группа «Основные».** Настройки программы группы «Основные» приведены на рисунке 8.1.

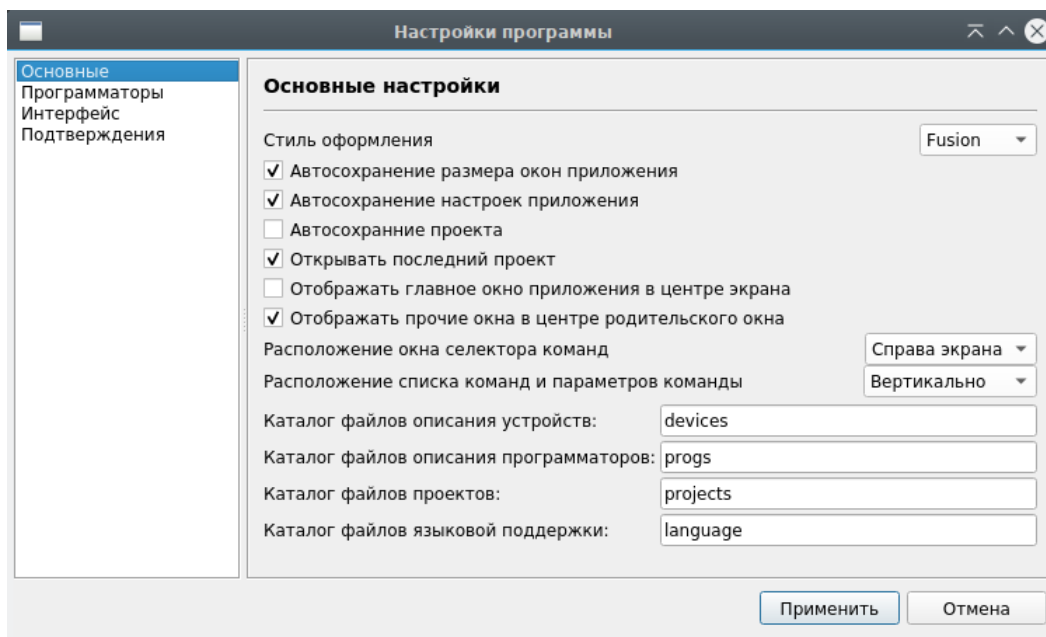


Рисунок 8.1. Настройки программы. Основные

8.3. **Настройки программы. Группа «Программаторы».** Настройки программы группы «Программаторы» приведены на рисунке 8.2.

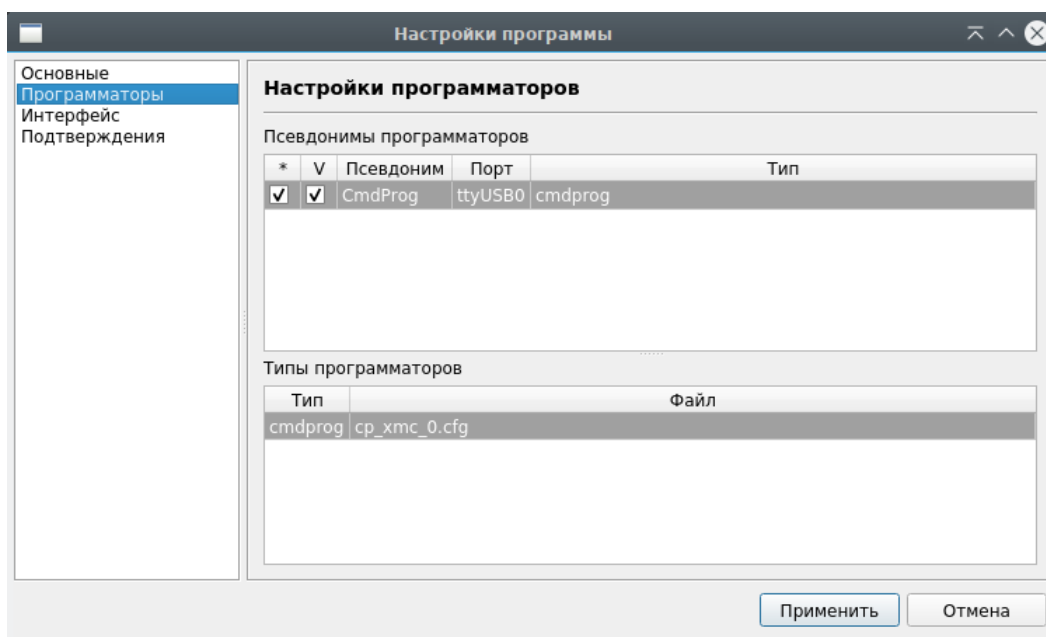


Рисунок 8.2. Настройки программы. Программаторы

Настройка программаторов в программе производится следующим образом:

- в панели «Типы программаторов» конфигурационному файлу описания программатора устанавливается имя типа программатора;
- в панели «Псевдонимы программаторов» устанавливается привязка псевдонима программатора, последовательного порта и типа программатора (установленного в панели «Типы программаторов»).

Такая настройка программаторов сделана для того, чтобы не менять в каждом проекте параметры команды «Set Prog». Достаточно изменить номер последовательного порта и тип программатора в панели «Псевдонимы программаторов».

В панели «Псевдонимы программаторов» имеются два столбца с именами «*» и «V». В столбце «*» выбирается программатор, который будет назначен по умолчанию при добавлении команды «Set Prog» в список команд. В столбце «V» выбираются программаторы, которые будут доступны при выборе в параметрах команды «Set Prog».

8.4. Настройки программы. Группа «Интерфейс». Настройки программы группы «Интерфейс» приведены на рисунке 8.3.

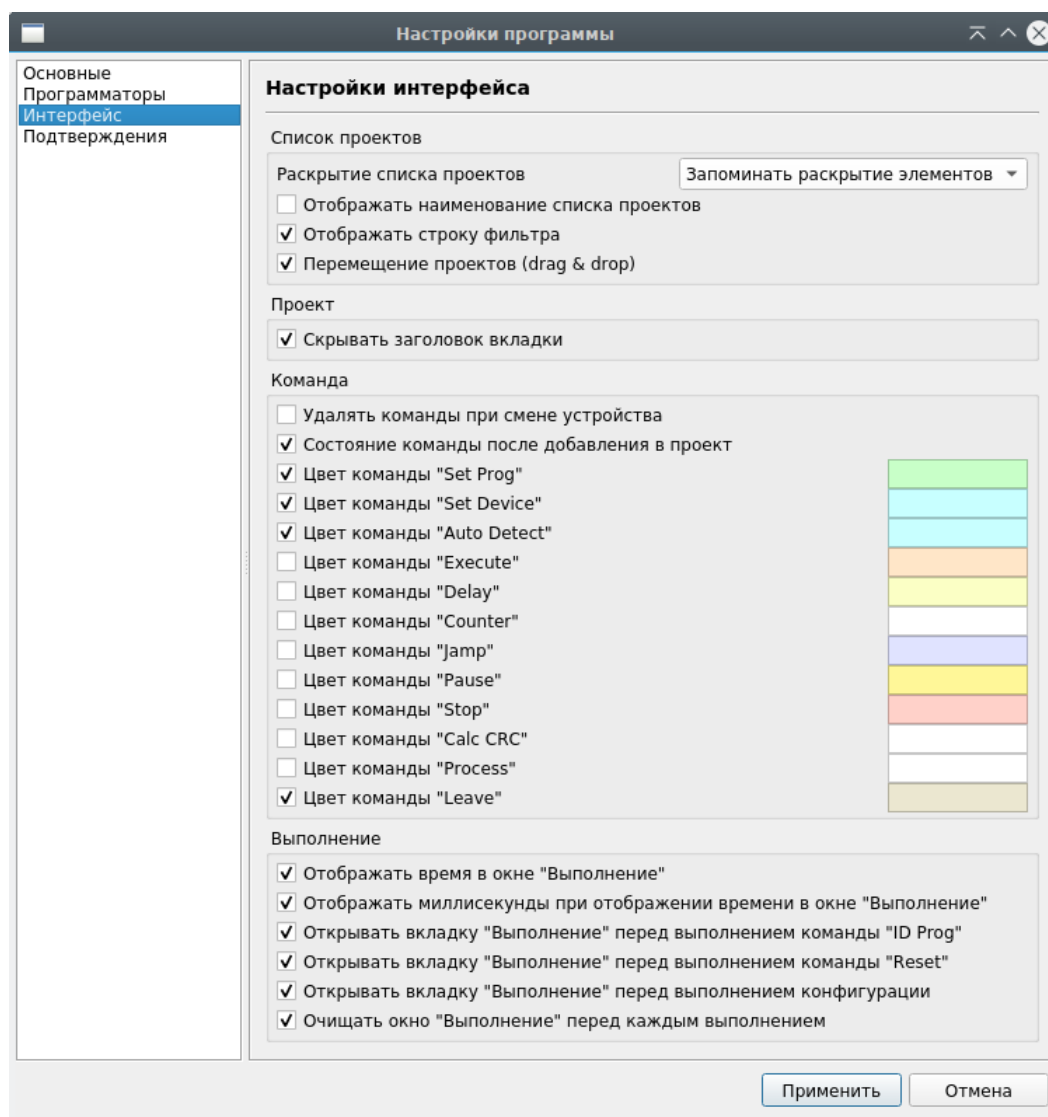


Рисунок 8.3. Настройки программы. Интерфейс

8.5. **Настройки программы. Группа «Подтверждения».** Настройки программы группы «Подтверждения» приведены на рисунке 8.4.

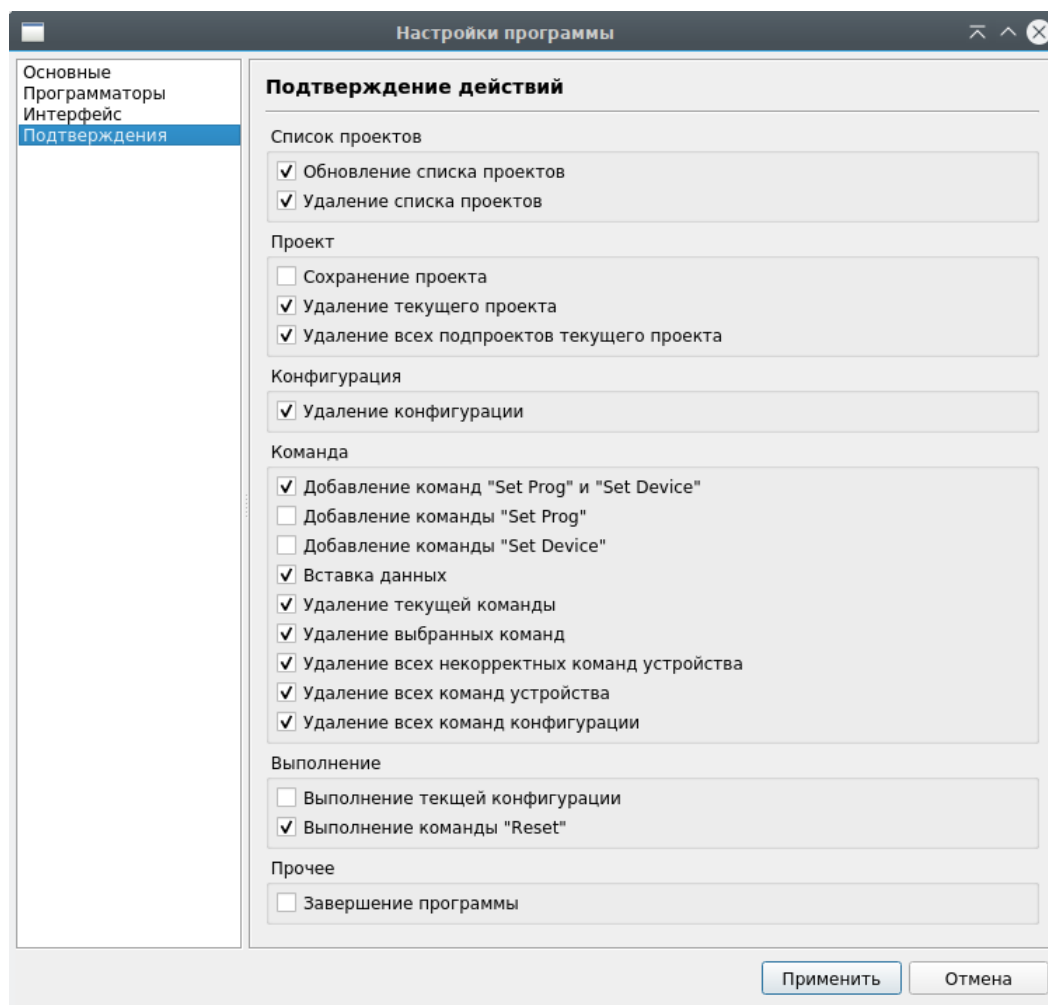


Рисунок 8.4. Настройки программы. Подтверждения

9. Аппаратная часть (базовая версия)

9.1. Для связи программатора с компьютером выбран интерфейс UART, включая его реализации через другие интерфейсы (USB, Ethernet). Выбор данного интерфейса освобождает от проблем с драйверами. Многие преобразователи USB-COM определяются в ОС Linux без дополнительных манипуляций.

Для увеличения скорости программирования преобразователи должны иметь достаточно большой буфер приёма/передачи. Максимальный размер пакета (точнее поле data), а также другие параметры программатора, задаются в файле конфигурации программатора, находящегося в директории «progs».

В базовой версии программатора связь с компьютером осуществляется через преобразователь USB-COM FT230XS (FTDI), что позволило поднять скорость обмена программатора с компьютером до 3 МБит/с.

9.2. Защита интерфейса USB от электростатических разрядов реализовано на сборке USBLC6.

9.3. Применение микросхемы ADuM121N1 (Analog Devices), помимо согласования уровней, позволило добавить гальваническую развязку.

9.4. В качестве основы программатора выбран микроконтроллер XMC1100T016X0064 из серии XMC1000 (Infineon). Использование данного микроконтроллера позволило запрограммировать устройства в диапазоне напряжений от 1.8 В до 5.0 В без дополнительного согласования уровней.

Основные характеристики микроконтроллера XMC1100T016X0064 (Infineon):

- объем Flash – 64 КБайт;
- объем RAM – 16 КБайт;
- максимальная скорость по интерфейсу UART (связь с компьютером) – 8 МБит/с;
- максимальная скорость по интерфейсу SPI – 16 Мбит/с;
- максимальная скорость по интерфейсу I2C – 400 КБит/с;
- напряжение питания – 1.8...5.0 В.

Микроконтроллер XMC1100T016X0064 можно заменить любым микроконтроллером из серии XMC1000 (требуется перекомпиляция).

Использование микроконтроллера серии XMC1400 (Infineon) в связке со скоростным преобразователем USB-COM позволит увеличить скорость обмена с компьютером до 12 Мбит/с. При этом сохраняется работа программатора в диапазоне напряжений питания 1.8...5.0 В без дополнительного согласования уровней.

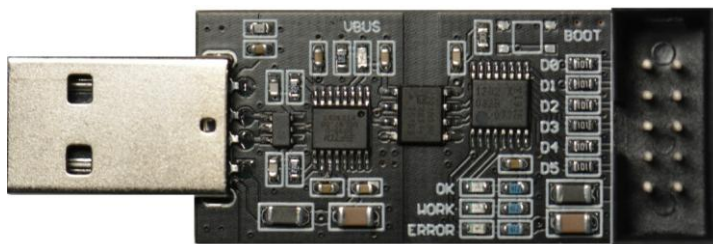


Рисунок 9.1. Внешний вид программатора CmdProg

9.5. На плате программатора расположены четыре индикатора «VBUS», «OK», «WORK», «ERROR», отображающие текущее состояние:

- «VBUS» (жёлтый) – индикатор питания шины USB;
- «OK» (зелёный) – процесс программирования завершился успешно, либо программатор включён и готов к работе;
- «WORK» (синий) – идёт процесс программирования;
- «ERROR» (красный) – процесс программирования завершился неудачно.

9.6. Программатор может быть реализован на любом микроконтроллере, имеющем необходимые аппаратные интерфейсы (включая аппаратный USB в режиме Virtual Com Port).

9.7. Использование программатора в ОС Windows. Для увеличения скорости обмена между компьютером и программатором, построенным на базе преобразователя USB-COM от компании FTDI, необходимо изменить параметры соответствующего последовательного порта, как показано на рисунке 9.2.

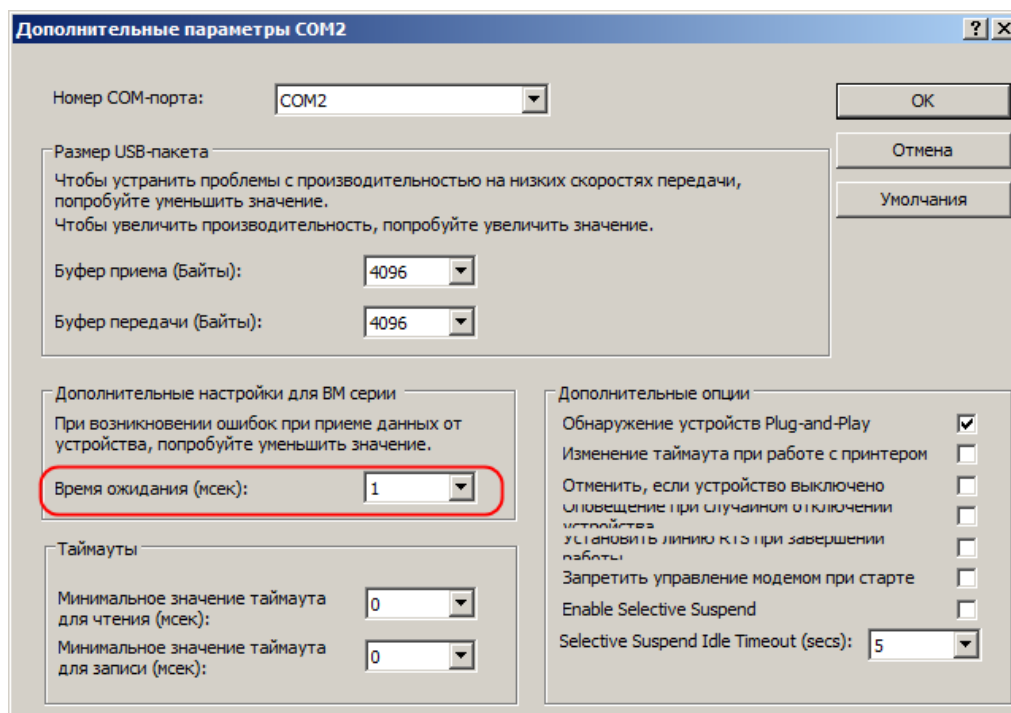


Рисунок 9.2. Параметры COM-порта в ОС Windows

9.8. Использование программатора в ОС Linux. Перед использованием последовательного порта в ОС Linux, необходимо включить пользователя в группу dialout.

Для увеличения скорости обмена между компьютером и программатором, построенным на базе преобразователя USB-COM от компании FTDI, необходимо в режиме суперпользователя изменить параметры соответствующего последовательного порта:

```
echo 1 > /sys/bus/usb-serial/devices/ttyUSB0/latency_timer
```