



Миландр 1986BE9X

Содержание

Миландр 1986BE9X.....	3
Введение.....	3
Настройка проекта для последующей генерации кода СИ.....	4
Настройка кодогенератора.....	5
Структура проекта Test_Driver.....	16
Удаленная отладка алгоритма.....	18
Совместная отладка алгоритмов на инструментальной машине и целевой системе в SimInTech.....	22

Миландр 1986BE9X

В данном руководстве описана технология разработки программного обеспечения для микроконтроллеров семейства 1986BE9X компании Миландр с использованием SimInTech. Данное руководство содержит инструкции по генерации кода на языке СИ из алгоритма управления, разработанного в SimInTech, для программирования микроконтроллеров семейства 1986BE9X. Автоматическая генерация кода на язык программирования СИ, из схемы алгоритма SimInTech, позволяет избежать ошибок, которые совершает программист при использовании “ручного программирования” и увеличить скорость разработки программного обеспечения.

Введение

После разработки и исследования проекта SimenTech на инструментальной машине, существует возможность удаленной отладки алгоритма на контроллере управления, для этого необходимо получить код алгоритма на языке СИ, откомпилировать его и перенести бинарный образ программы на контроллер управления. Подобный режим работы позволяет оценить корректность работы управляющего алгоритма на контроллере управления и, используя визуальные возможности SimenTech, отладить его. Обмен данными между схемой алгоритма в SimenTech и программой, работающей на управляющем контроллере происходит по последовательному интерфейсу RS232 с частотой шага интегрирования. В случае если алгоритм управления является частью проекта “пакет” (специальный тип проекта, где проекты *.prt объединенные в один проект пакет *.pak), то существует возможность совместной работы части проектов на инструментальной машине SimenTech и контроллере управления (модель объекта управления работает на инструментальной машине, а алгоритм управления на контроллере управления), для обмена сигналами между проектами работающими на инструментальной машине и контроллером управления используется последовательный интерфейс RS232 с частотой шага синхронизации проектов. Используя подобный режим работы получаем результаты полунатурного эксперимента и можем оценить потерю точности при работе алгоритма управления на микропроцессоре, благодаря отличиям в архитектурах процессоров на инструментальной машине SimenTech и управляющем контроллере (размеры вещественных данных, объем памяти и тд). Для демонстрации технологии разработки программного обеспечения, будет использован проект привода бесколлекторного двигателя постоянного тока, состоящий из двух проектов, объединённых в

пакет БДПТ.рак (проект бесколлекторного двигателя постоянного тока БДПТ.ptr и алгоритма управления Управление БДПТ.prt), проект находится в папке \SimInTech\Demo\Электроприводы\БДПТ\Пример_1\ и может быть скопирован в рабочую директорию. Для программирования контроллера управления будем использовать проект Управление БДПТ.prt, а в качестве контроллера управления будем использовать отладочную плату для процессора MDR32F2Q фирмы Миландр. Обмен данными между контроллером управления и инструментальной машиной будет происходить с использованием последовательного интерфейса RS232.

Настройка проекта для последующей генерации кода СИ

Для настройки генерации кода СИ проекта Управление БДПТ.prt необходимо:

1. Откройте проект, созданный в SimInTech, в котором содержится алгоритм, для которого необходимо сгенерировать код.

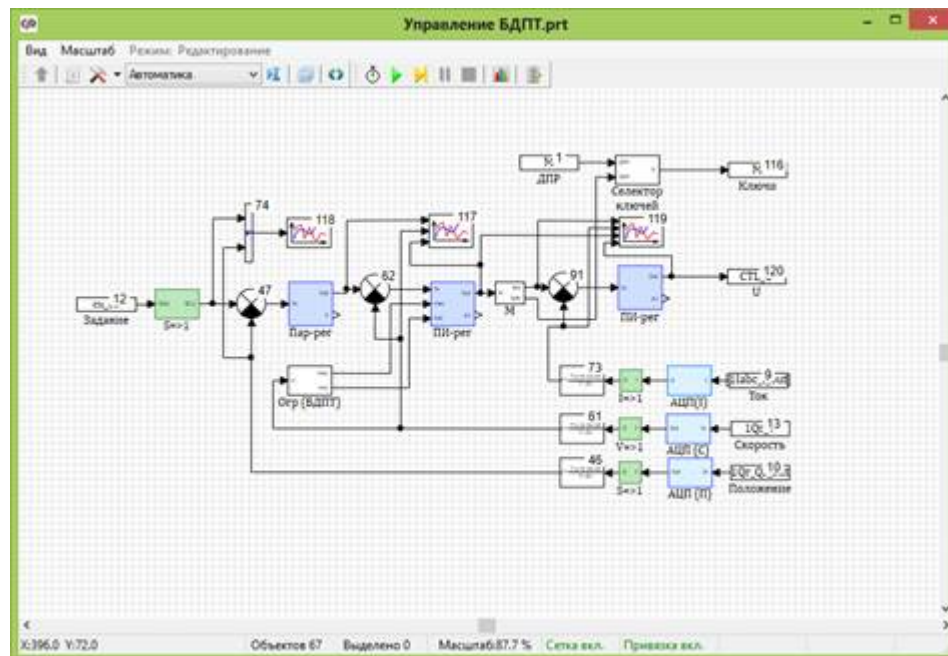


Рисунок 1. Окно проекта, созданного в SimInTech, с готовым алгоритмом

2. В окне проекта на панели кнопок нажмите на кнопку **Параметры расчёта**.
3. В появившемся окне параметров проекта перейдите на вкладку **Параметры расчёта**. Для свойства **Имя (имена) алгоритма** определите уникальное имя, которым будет обладать алгоритм, находящийся в данном проекте (*Рисунок 2. Редактирование имени алгоритма на вкладке «Параметры расчёта» окна параметров проекта* на странице 5).

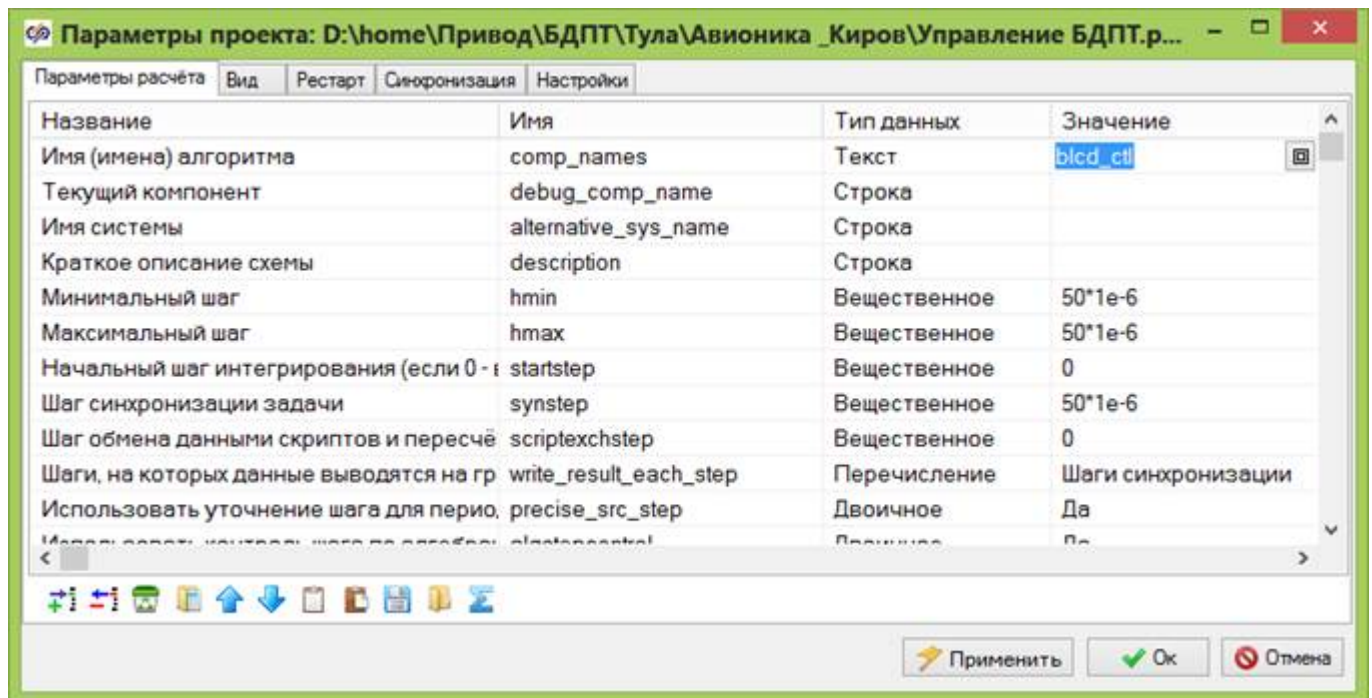


Рисунок 2. Редактирование имени алгоритма на вкладке «Параметры расчёта» окна параметров проекта

4. Закройте окно нажатием на кнопку **Ok**, сохраняя внесенные изменения.

Настройка кодогенератора

Для настройки кодогенератора выполните следующие действия:

1. В главном окне программы войдите в меню **Кодогенератор**, в котором выберите пункт **Кодогенератор СИ** (*Рисунок 3. Выбор кодогенератора в меню «Кодогенератор» главного окна программы* на странице 5). В результате перед вами появится окно кодогенератора (*Рисунок 4. Окно кодогенератора* на странице 6).

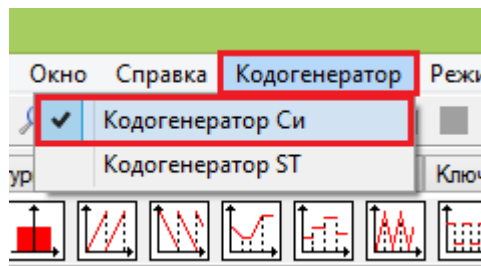


Рисунок 3. Выбор кодогенератора в меню «Кодогенератор» главного окна программы

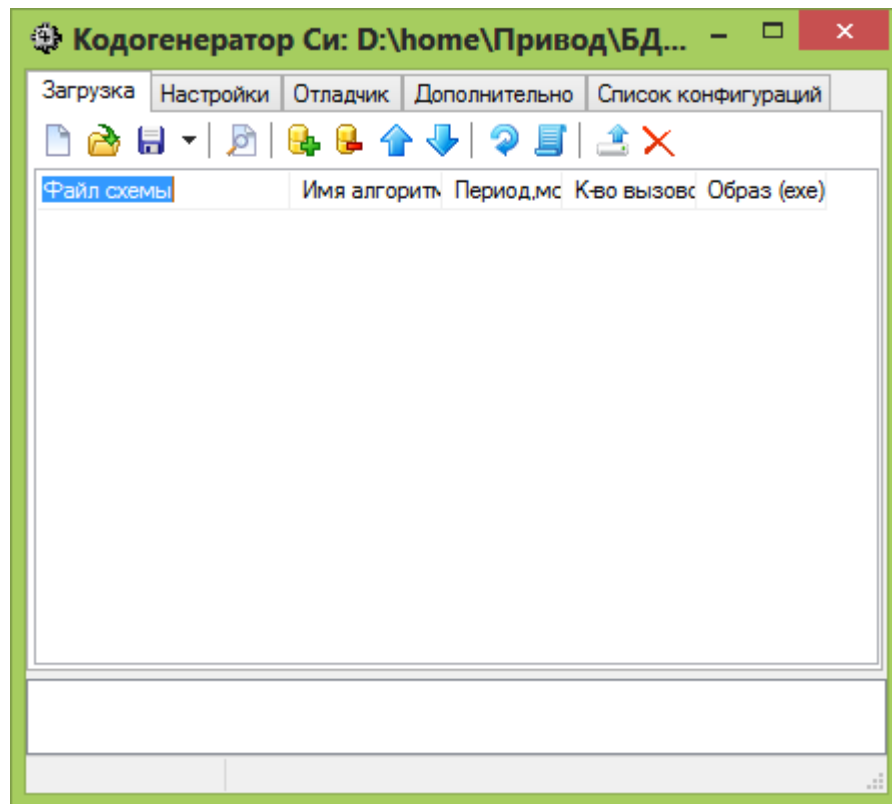


Рисунок 4. Окно кодогенератора

Данное окно можно разделить на четыре области [Рисунок 5. Области окна кодогенератора](#) на странице 7:

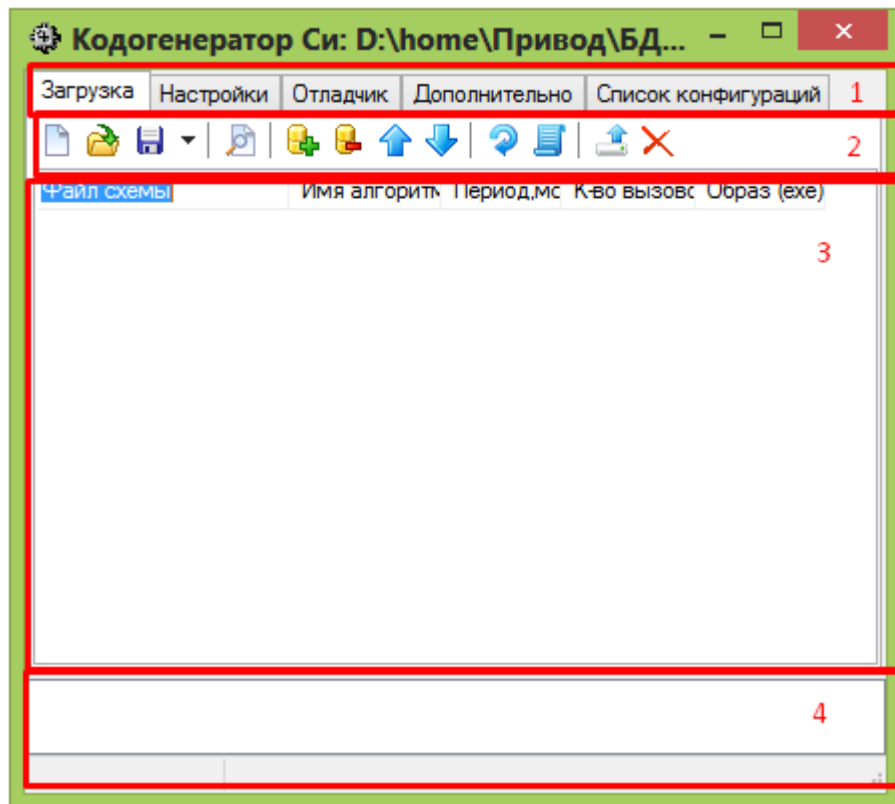


Рисунок 5. Области окна кодогенератора

- 1 область – область меню;
- 2 область – панель кнопок, в данной области содержатся кнопки для сохранения файла пакета проектов, добавления, удаления и расположения проектов в списке, управления расчетом и настройки пакета;
- 3 область – список проектов. В виде таблицы содержит проекты, которые будут сгенерены;
- 4 область – окно сообщений. В данной области отображаются диагностические сообщения о загрузке проектов, входящих в пакет, их инициализации, расчете пакета, загрузке/сохранение рестартов, остановке пакета проектов и пр.

Панель кнопок содержит следующие кнопки:

- Новая конфигурация – открывает пустую конфигурацию с начальными настройками.
- Загрузить конфигурацию – открывает окно, в котором можно выбрать, а затем загрузить ранее сохраненную конфигурацию.
- Сохранить конфигурацию – сохраняет текущую конфигурацию с именем, которое определил пользователь. Файл конфигурации после сохранения имеет расширение .cfg.

- Открыть выделенные – открывает выделенные пользователем в таблице проекты.
 - Добавить файлы – открывает окно выбора проекта для добавления в текущую загруженную конфигурацию.
 - Удалить файлы – удаляет выделенные пользователем проекты из конфигурации.
 - Сдвинуть вверх – перемещает выбранный пользователем в таблице проект на одну позицию выше.
 - Сдвинуть вниз – перемещает выбранный пользователем в таблице проект на одну позицию ниже.
 - Пересобрать модули и конфигурацию – не используется при генерации кода для микроконтроллера.
 - Пересобрать только конфигурацию загрузки – не используется при генерации кода для микроконтроллера.
 - Скопировать на целевую систему – не используется при генерации кода для микроконтроллера.
 - Очистить целевую систему – не используется при генерации кода для микроконтроллера.
2. В появившемся окне на панели кнопок нажмите на кнопку **Добавить файлы** (*Рисунок 6. Кнопка добавления файлов на странице 9*).

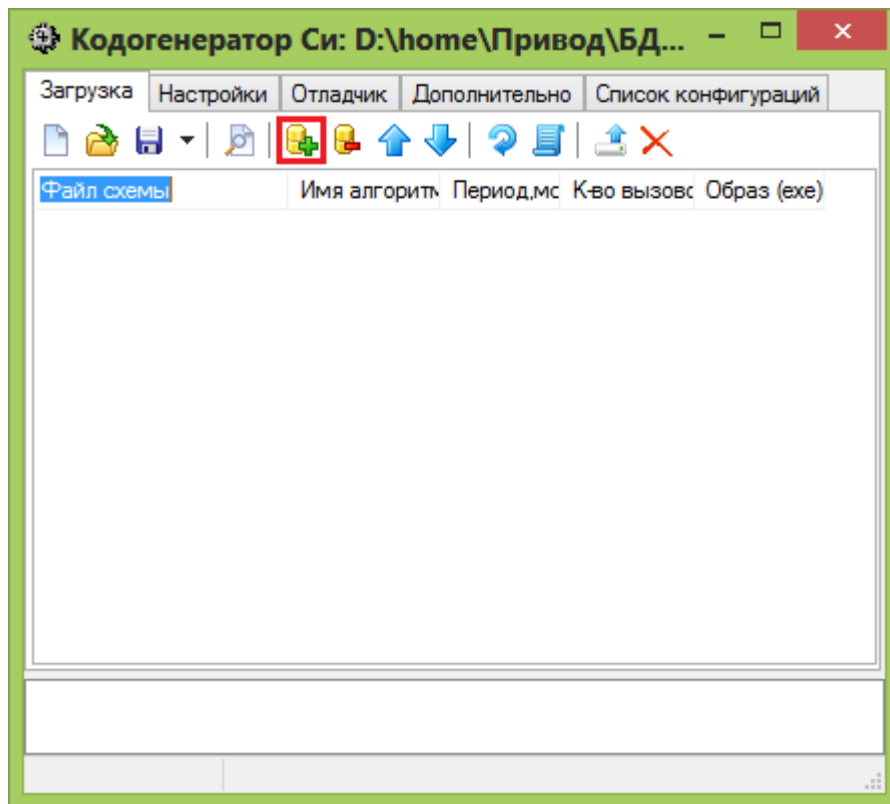


Рисунок 6. Кнопка добавления файлов

В результате откроется окно добавления проекта.

3. В появившемся окне добавления проекта выберите проект и нажмите на кнопку «Открыть».

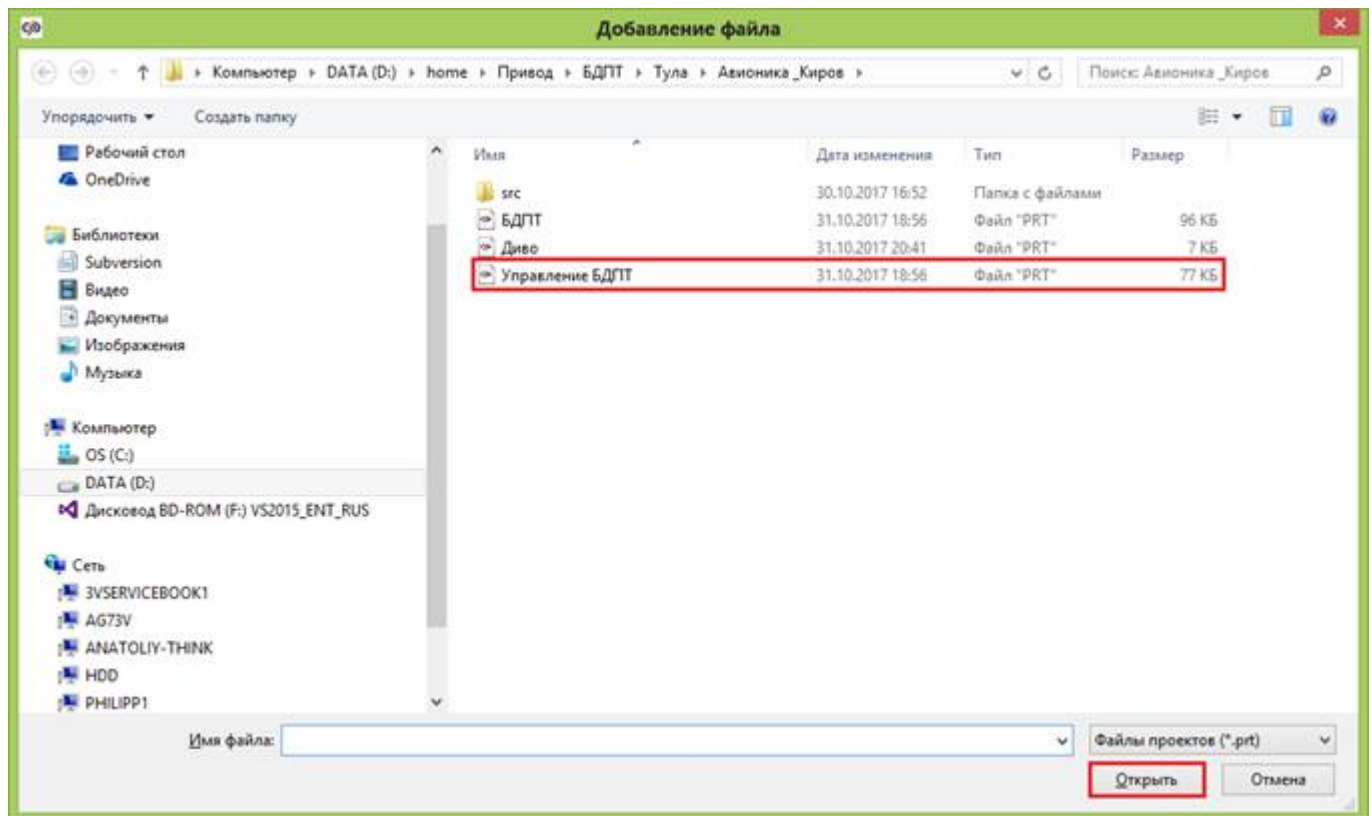


Рисунок 7. Окно открытия файла

4. Добавьте проект в таблицу проектов. Обратите внимание, что при добавлении проекта значения полей **Период, мс** и **К-во вызовов** заполняются автоматически и не имеют значения для дальнейшей работы, эти поля используются при генерации кода для контроллеров управления использующих операционные системы КПДА и Linux (ARM) и означают они период выполнения алгоритма в миллисекундах и количество вызовов на каждом такте работы алгоритма.

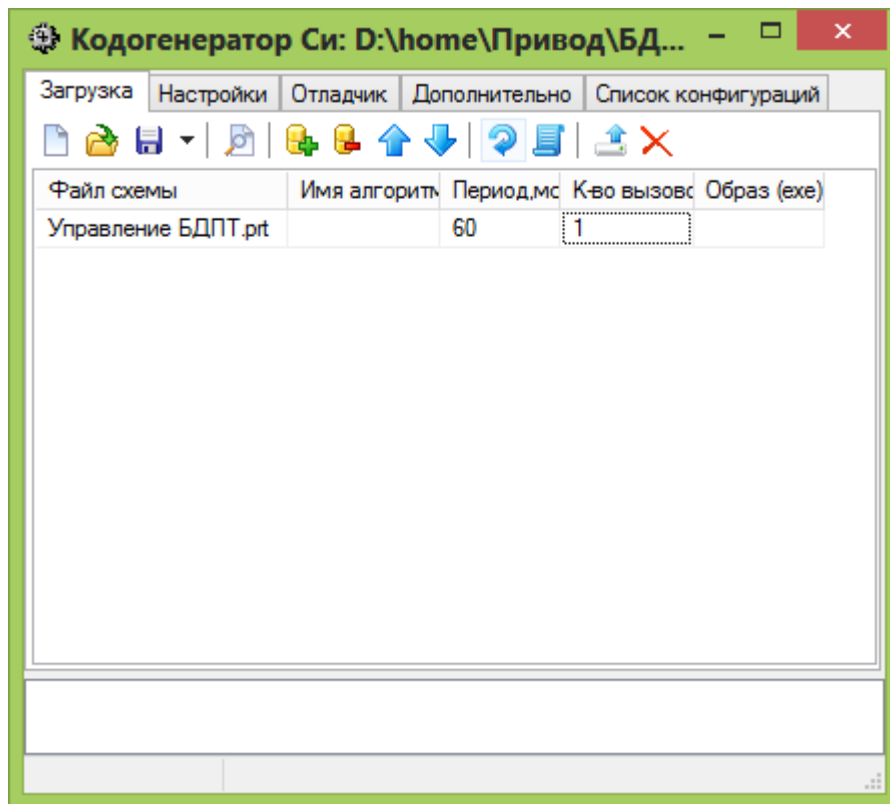


Рисунок 8. Окно кодогенератора

5. Выбранные настройки необходимо сохранить в файле конфигураций. Для этого перейдите на вкладку **Загрузка** и нажмите на кнопку **Сохранить как** ([Рисунок 9. Кнопка сохранения конфигурации](#) на странице 12). Файл конфигураций должен быть сохранен в рабочей директории проекта ([Рисунок 10. Сохранение файла конфигурации](#) на странице 12), потому что при задании в настройках относительного пути сохранения результатов кодогенерации, результирующая директория исходных кодов программы будет находится относительно директории в которой сохранен файл конфигурации.

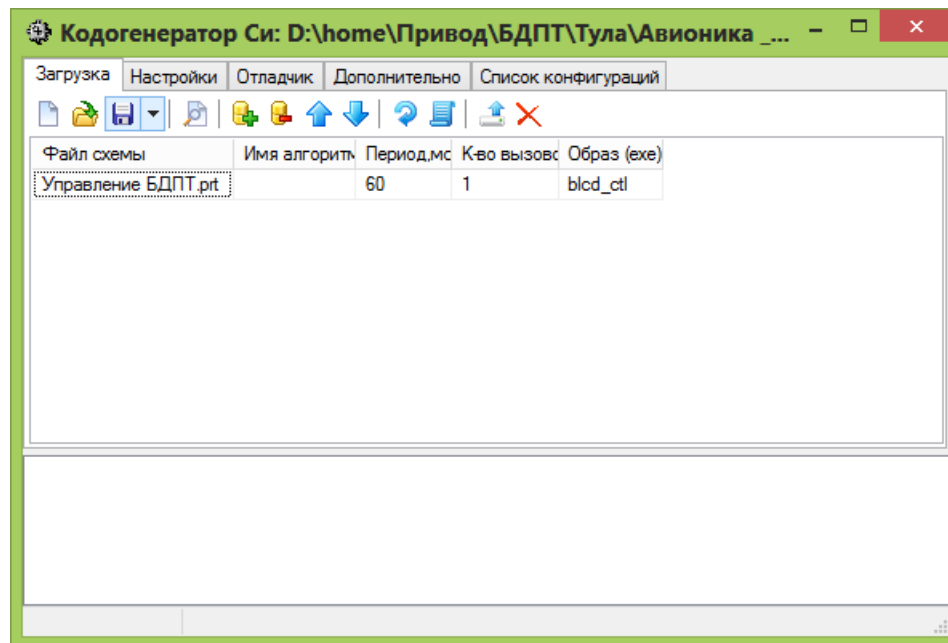


Рисунок 9. Кнопка сохранения конфигурации

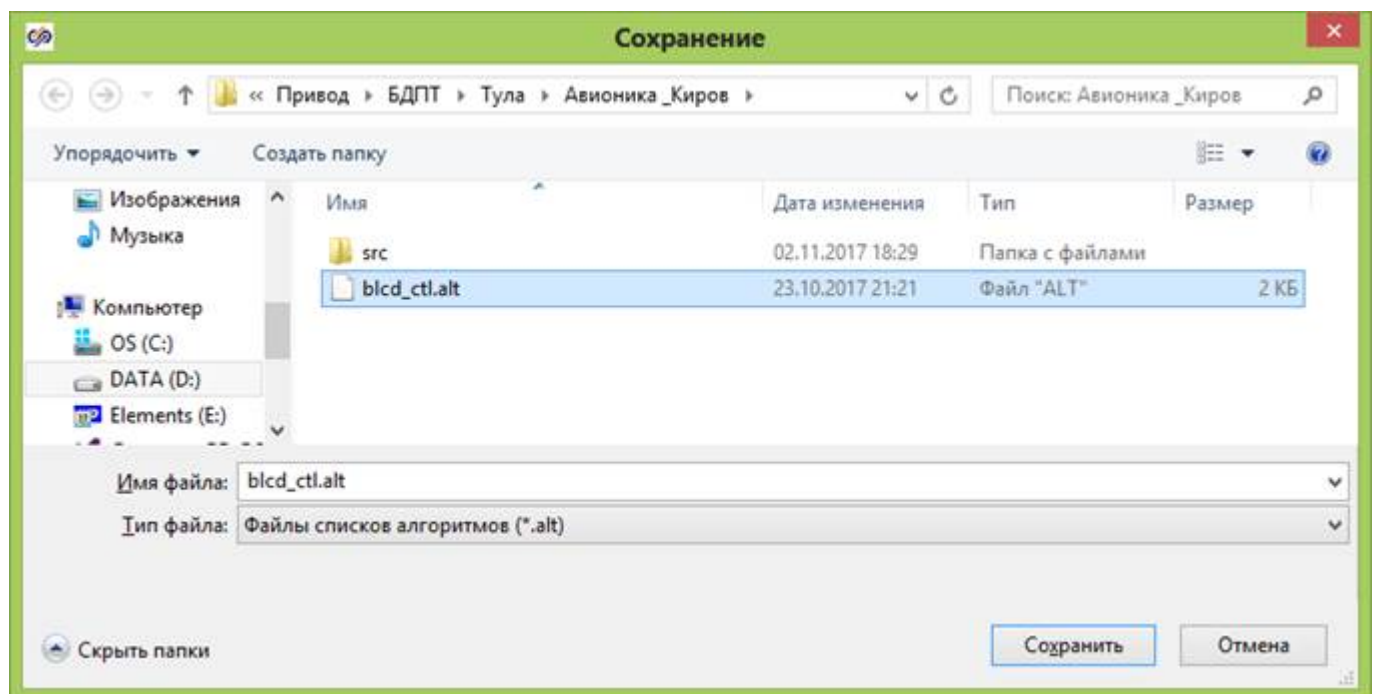


Рисунок 10. Сохранение файла конфигурации

6. Перейдите на вкладку **Настройки**. На данной вкладке необходимо заполнить следующие поля ([Рисунок 11. Настройки кодогенератора](#) на странице 13):

- Директория шаблона кода – директория, в которой расположен шаблон кода;
- Директория исходников – директория, в которой будут сохранены файлы, полученные в результате работы кодогенератора.

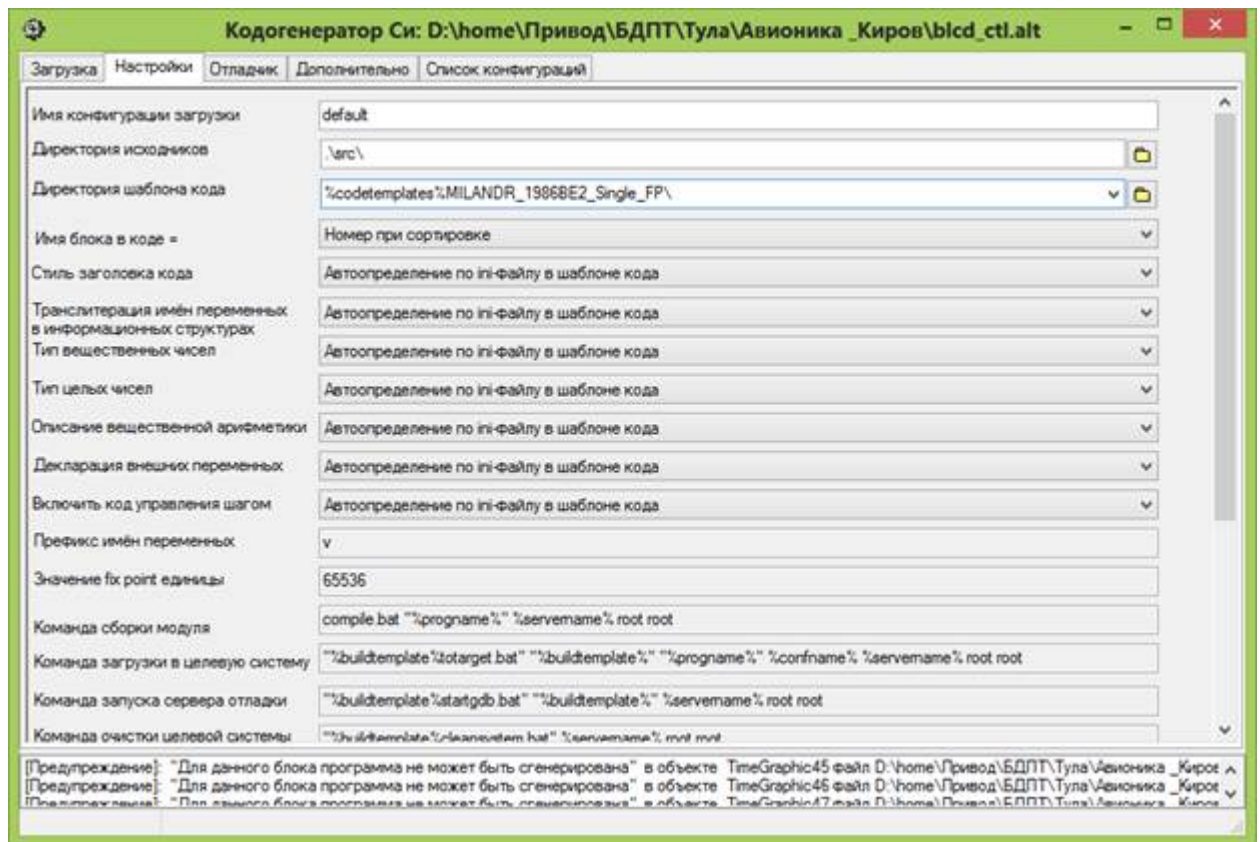


Рисунок 11. Настройки кодогенератора

В директории шаблонов, выберите шаблон для генерации кода под микроконтроллер “Миландр” - %codetemplates%MILANDR_1986BE2_Single_FP \. Укажите директорию для хранения результатов кодогенерации .\src\ (можно задать директорию относительно рабочей директории проекта или можно задать абсолютный путь).

7. Перейдите на вкладку **Загрузка** и нажмите на кнопку **Пересобрать модули и конфигурацию**. В окно сообщений будет выдана информация о результатах кодогенерации см.([Рисунок 11. Настройки кодогенератора](#) на странице 13, [Рисунок 14. Сообщения кодогенератора](#) на странице 16).

Если генерация кода была завершена успешно, то в результате кодогенерации в директории \src\<blcd_ctl>\, где blcd_ctl – это имя алгоритма, которое мы указали при настройке параметров проекта, создастся проект см. ([Рисунок 13. Панка проекта для Keil uVision](#) на странице 15), включающий в себя исходные СИ коды алгоритма (наличие готового проекта позволит быстрее отладить алгоритм на контроллере управления):

- Test_Driver.uvproj (для IDE Keil uVision4, включающий в себя файлы алгоритмов на языке СИ и драйвера необходимые для работы с устройствами ввода

вывода, отладочной платы), после компиляции проекта бинарный образ алгоритма можно транслировать на контроллер управления.

Дополнительно в директории см. (Рисунок 12. Состав папки src на странице 14) `.\src` расположены файлы `blcd_ctl.extvars.table` и `blcd_ctl.intvars.table`, они будут необходимы для удаленной отладки алгоритма на целевой системе, файлы `blcd_ctl.inputs` и `blcd_ctl.outputs` содержат списки сигналов входных и выходных, соответственно, файл `default.conf` не используется при данном варианте генерации кода.

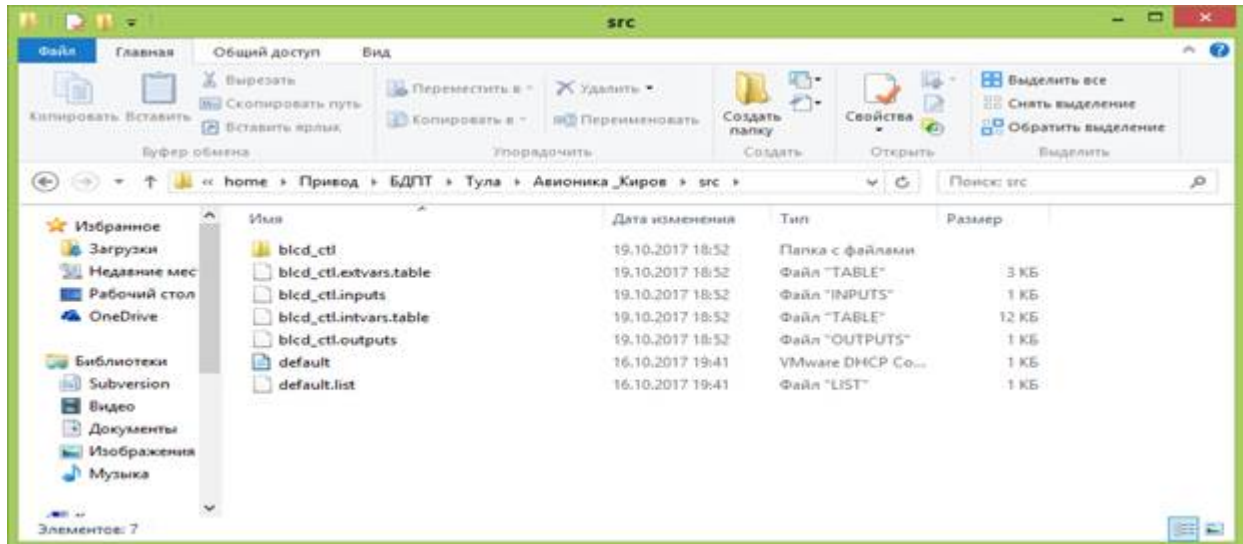


Рисунок 12. Состав папки src

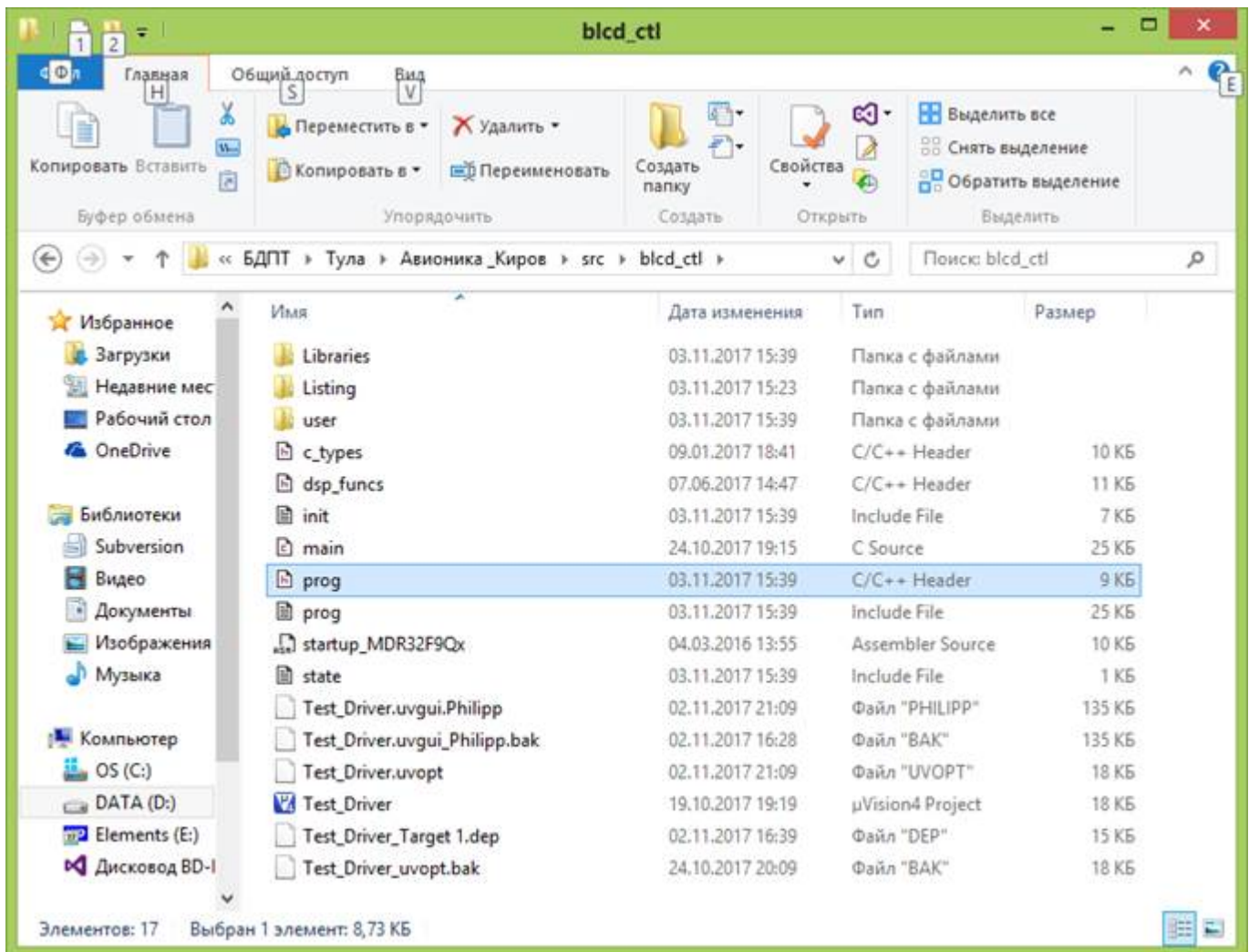


Рисунок 13. Папка проекта для Keil uVision

Стоит отметить, что генерацию кода можно произвести двумя способами приведенных ниже:

- 1 способ – воспользоваться кнопкой **Пересобрать модули и конфигурацию**, описанный выше;
- 2 способ – воспользоваться меню **Инструменты** главного окна программы, в котором выбрать пункт **Сгенерировать программу**, отличие от первого способа заключается в том, что не будет создан файл `default.conf` и результаты генерации кода будут выведены в окно сообщения проекта.

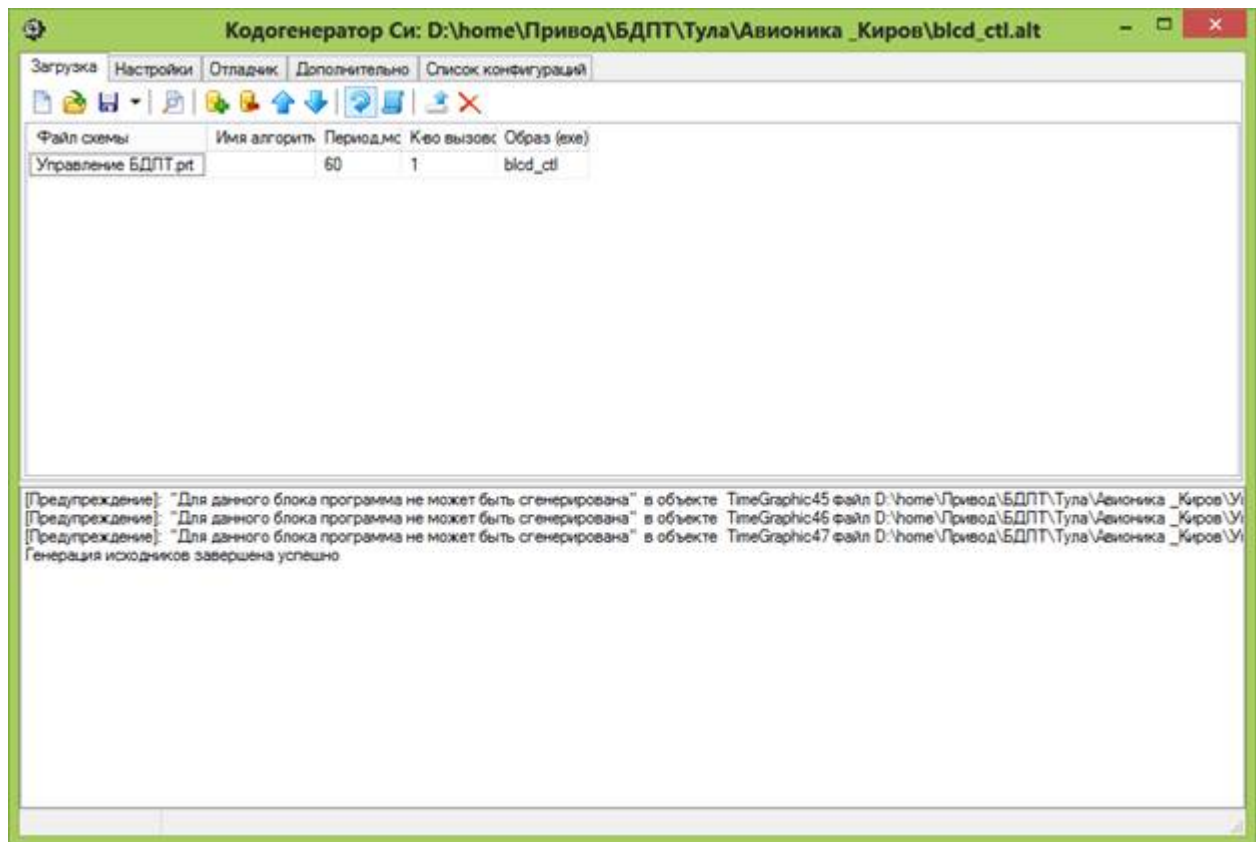


Рисунок 14. Сообщения кодогенератора

Структура проекта Test_Driver

Проект `Test_Driver.uvproj`, который является результатом работы генератора кода, расположен в директории `/src/blcd_ctl/Test_Driver`, где `/src` директория исходников, указанная в настройках кодогенератора, `/blcd_ctl` поддиректория в директории исходников, имя которой определено в поле "Имя алгоритма", во вкладке параметры расчета проекта. Проект может быть изменен пользователем. После сборки проекта и трансляции бинарного образа на целевую систему, алгоритм будет выполняться с определенным временным тактом. Структура проекта:

- Модуль инициализации `startup_MDR32F9Qx.s` для выбранной микросхемы.
- Файл `main.c` включает исходные тексты алгоритмов и периодический цикл их исполнения.
- `prog.h` (содержит определения основных типов используемых переменных и констант и определения глобальных переменных, а также определения подпрограмм обработчиков прерываний от периферийных устройств), `init.inc` (инициализация переменных начальными значениями, а так же вызовы инициализирующих функций периферийных устройств библиотеки работы с

устройствами ввода/вывода), state.inc, prog.inc сгенерированные тексты алгоритмов на языке СИ.

- Файлы драйверов устройств ввода/вывода расположенные в директории / Libraries.
- Инклюд-файлы содержащие библиотечные функции работающие с устройствами ввода вывода, к ним происходит обращение из визуальных блоков библиотеки работы с устройствами ввода вывода.

На рисунке представлен основной цикл выполнения алгоритма функции main, перед его выполнением происходит вызов функций настройки генератора тактовой частоты и инициализации переменных алгоритма INIT_FUNC(). Цикл реализован в while последовательным вызовом функций RUN_FUNC() и STATE_FUNC(), после вызова которых происходит ожидание истечения временной задержки или прохождения периода ШИМ. Пользователь самостоятельно может выбрать подходящий вариант, убрав комментарии соответствующих временных функций (Delay_ms() или DelayTimer_pwm ()) в зависимости от решаемой задачи.

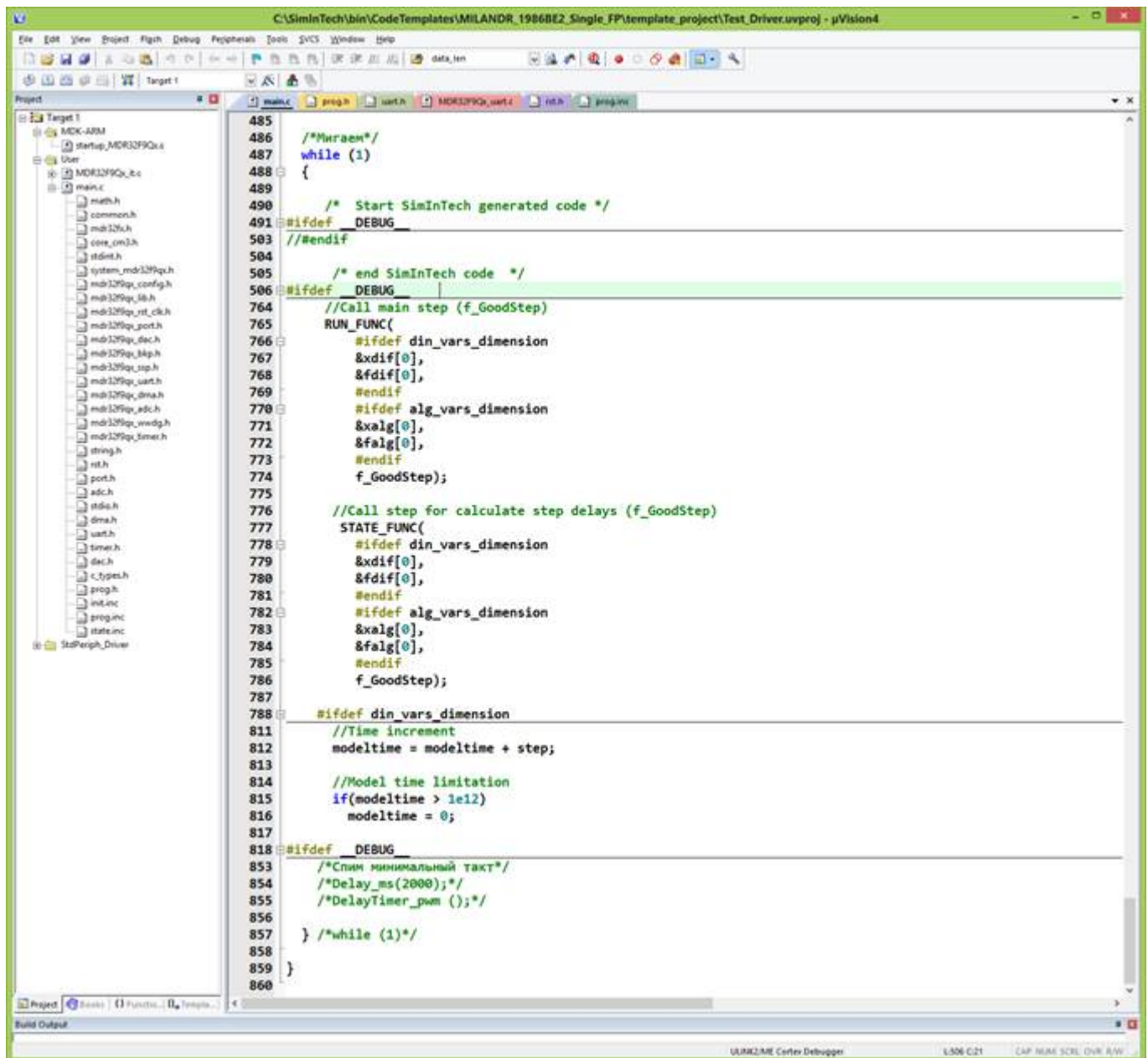


Рисунок 15. Структура проекта в Keil uVision

Перед основной функцией `main()` определены функции используемые в основном цикле алгоритма и подключены инклюд-файлы, содержащие реализации функций работы с устройствами ввода/вывода, их будут использовать библиотечные блоки работы с устройствами ввода/вывода.

Удаленная отладка алгоритма

Существует возможность удаленной отладки алгоритма на целевой системе. Используя удаленный режим работы можно оценить работу алгоритма управления (скорость

исполнения, точность регулирования) на процессоре Миландр. При использовании удаленного режима на схему алгоритма в SimenTech будут передаваться значения выходных сигналов, динамические переменные, временные задержки, состояния триггеров, константы, локальные переменные будут вычисляться на схеме, по исходным данным. Существует возможность изменять константные переменные на схеме алгоритма и значения входных сигналов, они будут транслироваться на целевую систему. Обмен между моделью алгоритма в SimenTech и программой, работающей на целевой системе происходит с использованием протокола RS232.

Для этого необходимо выполнить следующие шаги:

1. Провести генерацию кода алгоритма в соответствии с приведенными выше рекомендациями.
2. Собрать бинарный образ алгоритма используя проект Test_Driver, в среде разработки Keil uVision4, предварительно установив в main.c режим отладки (`#define __DEBUG__`) или поместив на рабочее окно проекта библиотечный блок DEBUG, который после кодогенерации установит режим отладки (`#define __DEBUG__`).
3. Используя среду разработки Keil uVision4 загрузить бинарный образ алгоритма на целевую систему.
4. На вкладке “Отладчик” Кодогенератора Си установить параметры соединения с целевой системой:
 - Параметры соединения (хост:порт), рекомендованные значения COM1:115200 (COM1 обозначения интерфейса последовательного доступа на инструментальной машине, 115200 скорость обмена между инструментальной машиной и целевой системой).
 - Имя DLL удаленного доступа `rsdbg.dll`.

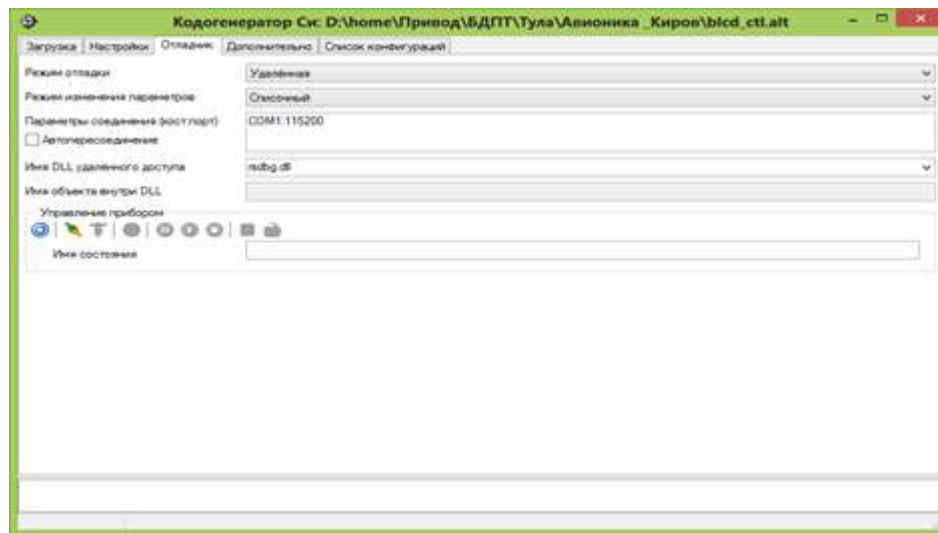


Рисунок 16. Окно кодогенератора

Перед началом удаленной отладкой алгоритма необходимо на главной панели SimenTech установить **Режим отладки: Удаленный** ([Рисунок 17. Включение удаленной отладки в главном меню](#) на странице 20) или зайти во вкладку параметры расчета алгоритма и указать **Режим отладки: Удаленный** ([Рисунок 18. Включение удаленной отладки в настройках проекта](#) на странице 21).

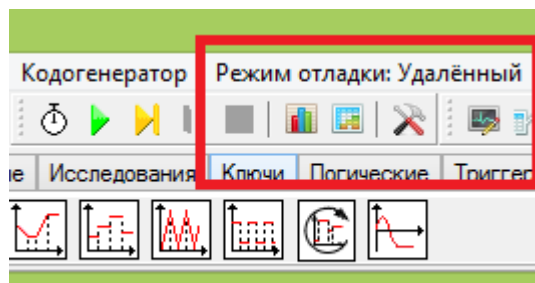


Рисунок 17. Включение удаленной отладки в главном меню

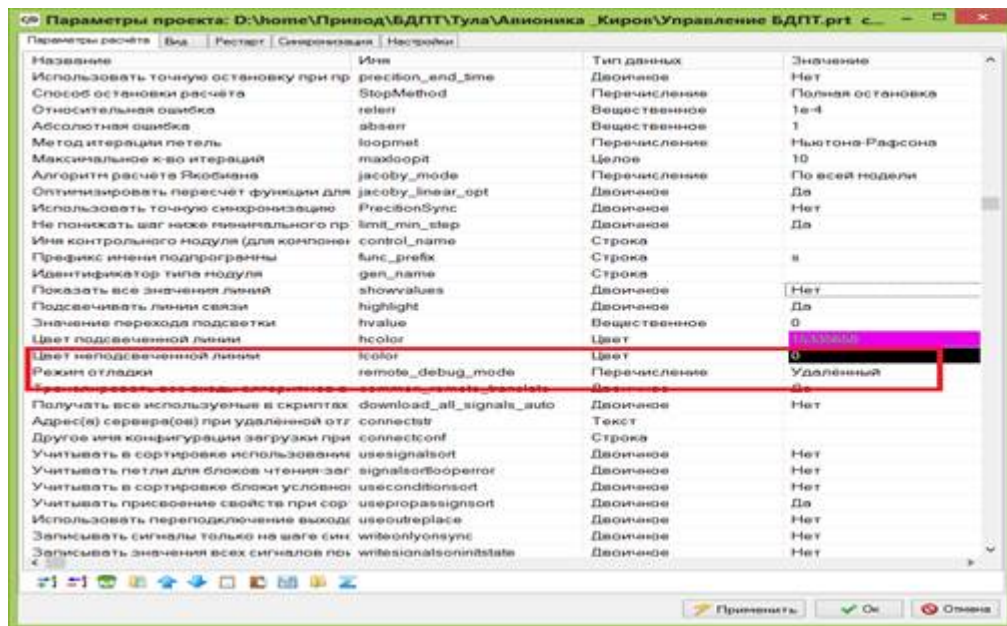


Рисунок 18. Включение удаленной отладки в настройках проекта

Запустить проект на выполнение, если проект успешно выполняется, то в левом нижнем углу будет происходить изменение расчетного времени с заданным шагом интегрирования, а заголовок окна будет содержать параметры соединения с инструментальной машиной.

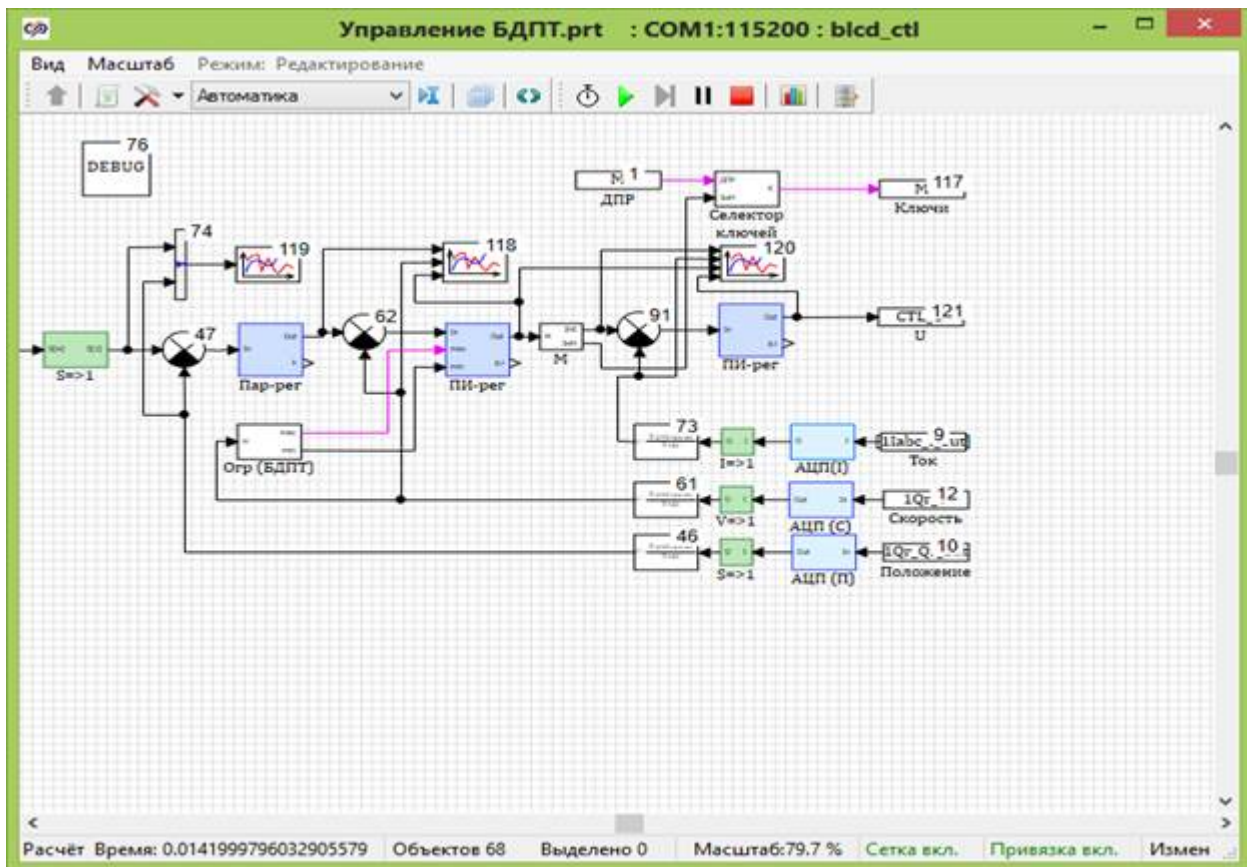


Рисунок 19. Окно проекта при удаленной отладке

Внимание! При перезапуске удаленной отладки проекта, необходимо перезагрузить целевую систему!

Совместная отладка алгоритмов на инструментальной машине и целевой системе в SimInTech

После математического моделирования объекта управления и алгоритма на инструментальной машине существует возможность перенести работу части алгоритмов на контроллер управления и отладить совместную работу алгоритмов на инструментальной машине и контроллере управления. Обмен сигналами между моделью объекта управления и алгоритмом управления, работающим на целевой системе происходит по последовательному интерфейсу RS232 с заданной скоростью обмена. Цель подобного режима работы заключается в проверке работы алгоритма управления на контроллере управления.

Для этого необходимо выполнить следующие шаги:

1. Объект управления и алгоритмы управления должны быть объединены в пакет проектов (БДПТ.pak)

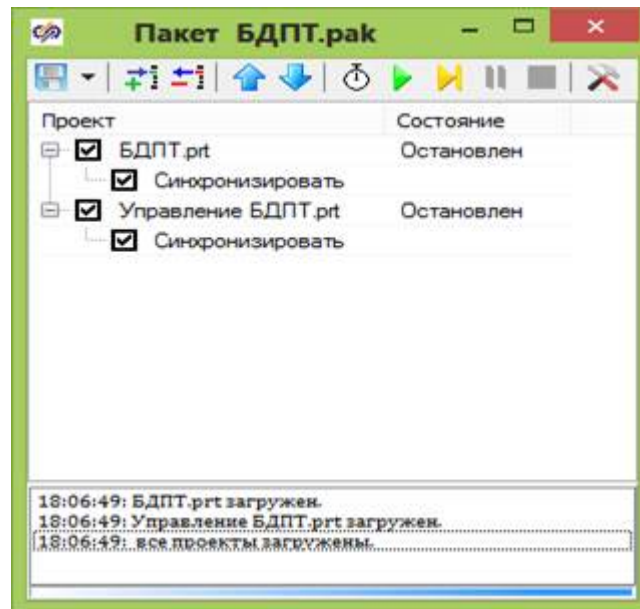


Рисунок 20. Менеджер проектов

2. Во вкладке “параметры расчета” проекта БДПТ.ptr (Рисунок 21. Параметры расчета на странице 24) модели объекта управления и алгоритмов управления установить шаг синхронизации – шаг, с которым будет происходить обмен сигналами между проектами в процессе моделирования. Параметр «Записывать сигналы только на шаге синхронизации в синхронизирующем потоке: да», параметр “Шаги на которых выводятся на графики: все шаги” этот параметр устанавливается в случае если мы хотим видеть графики с шагом равным интегрированию в ином случае, графики будут выводиться с установленным шагом синхронизации проекта.

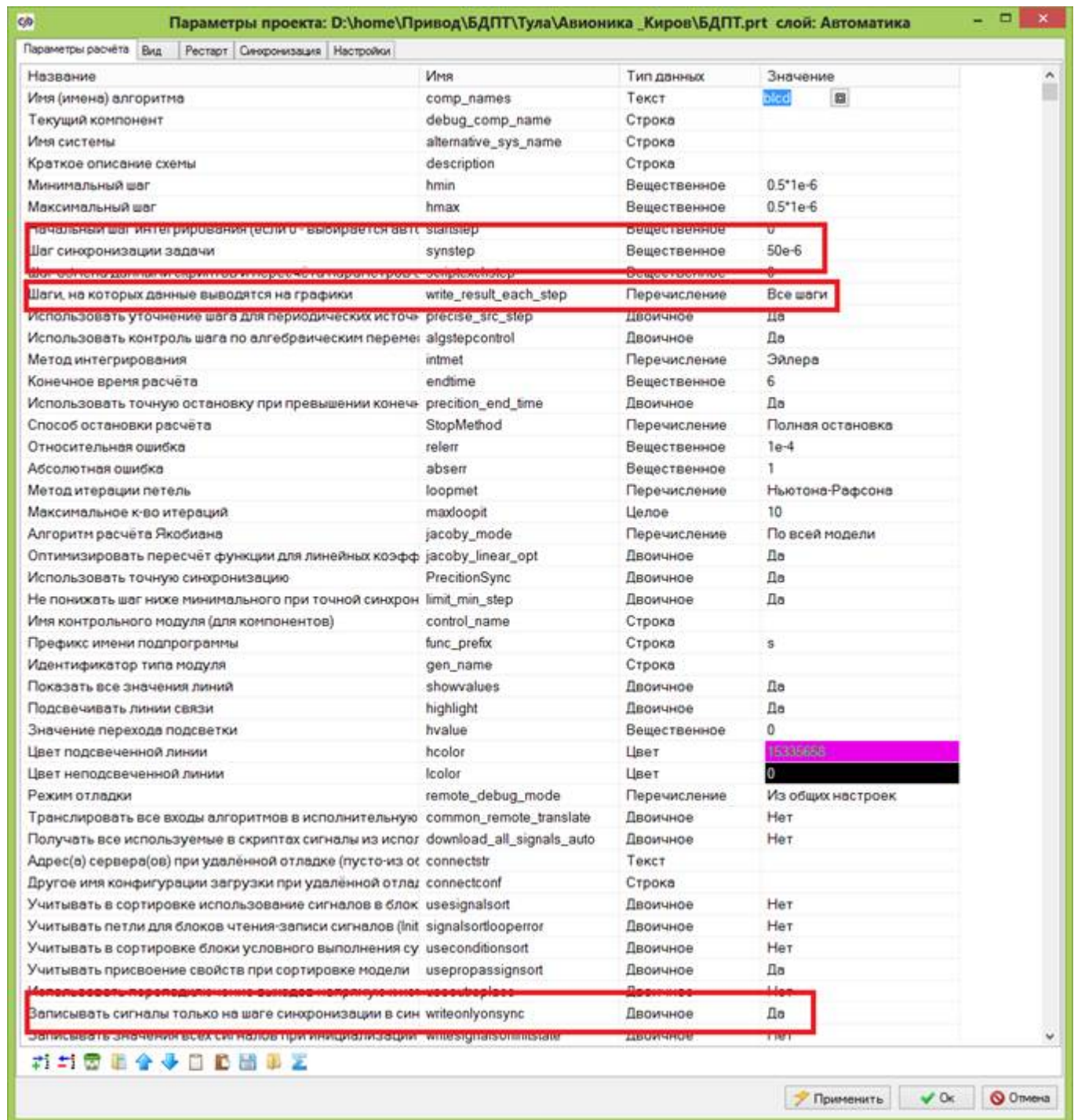


Рисунок 21. Параметры расчета

3. В “Параметры расчета” алгоритма управления Управление БДПТ.prt (*Рисунок 22. Параметры расчета схемы управления* на странице 25) дополнительно установить параметр “Режим отладки: Удаленный”, а параметр “Транслировать все входы алгоритмов в исполнительную систему и все выходы из неё: да”, установить шаг синхронизации проектов, такой же, как и у модели объекта управления.

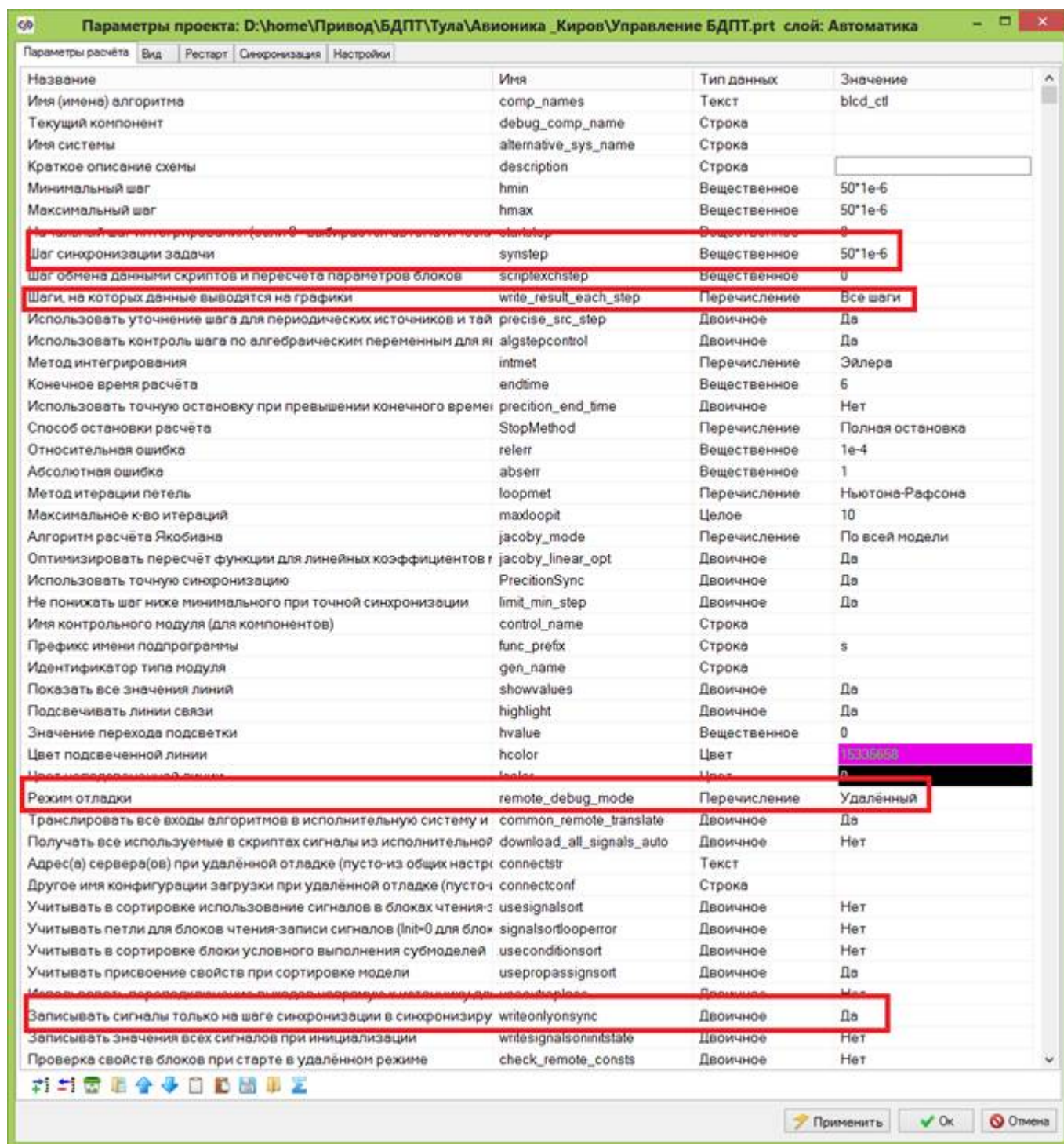


Рисунок 22. Параметры расчёта схемы управления

4. Режим отладки установить как “Локальный” (Рисунок 23. Режим отладки для схемы управления на странице 26) в главном меню SimenTech, таким образом модель объекта управления будет моделироваться на инструментальной машине (локально), а алгоритм управления будет работать на целевой системе, обмен

данными между моделью объекта управления и алгоритмом будет происходить с шагом синхронизации установленным во вкладке “Параметры расчета”.

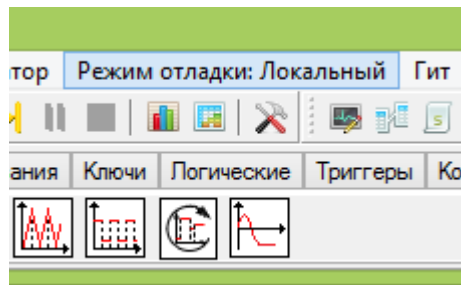


Рисунок 23. Режим отладки для схемы управления

5. Настроить генерацию кода и сгенерировать программу для проекта Управление БДПТ.prt в соответствии с приведенными выше инструкциями, а затем загрузить ее на контроллер управления, осуществить перезапуск контроллера управления.
6. После настроек проектов, входящих в пакет нажимаем кнопку “Пуск” ([Рисунок 24. Запуск пакета](#) на странице 26). Время работы пакета может происходить медленнее реального времени, это зависит от скорости моделирования объекта управления на инструментальной (локальный режим работы) и от скорости обмена между инструментальной машиной и целевой системой по RS232.

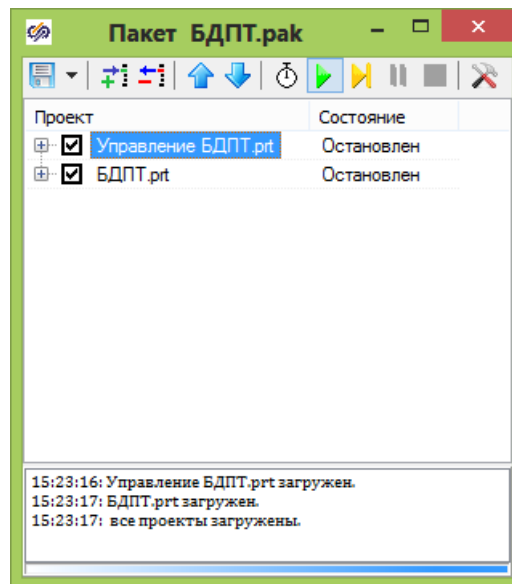


Рисунок 24. Запуск пакета