

Лабораторная работа № 1

Начальное изучение инструментальной системы ISaGRAF

Цель работы – ознакомление со средой инструментальной системы ISaGRAF и приобретение практических навыков программирования логических контроллеров на базе соответствующих средств инструментальной системы ISaGRAF

Методические указания

1 Краткое теоретическое введение

1.1 Архитектура проекта

Проект ISaGRAF разделен на несколько программных объектов, называемых программами. Связь между программами проекта имеет древовидную архитектуру. Программы могут быть разработаны на следующих графических и текстовых языках: SFC, FBD, LD, ST или IL.

1.1.1 Программы

Программа - это логический программный объект, который описывает взаимодействие между переменными процесса. Программы описывают либо последовательные, либо циклические операции.

Циклические программы выполняются на каждом цикле системы исполнения. Исполнение последовательных программ подчиняется динамическим правилам языка SFC.

Программы связаны между собой в виде иерархического дерева. Программы, расположенные на вершине дерева, активизируются системой. Подпрограммы (более низкий уровень иерархии) активизируются своими родительскими программами.

Программа может быть написана на следующих графических или текстовых языках:

- Язык последовательных функциональных схем (SFC) для программирования высокого уровня.
- Язык функциональных блочных диаграмм (FBD) для сложных циклических операций.
- Язык релейных диаграмм (LD) для булевых операций.
- Язык структурированного текста (ST) для любых циклических операций.
- Язык инструкций (IL) для операций низкого уровня.

1.1.2 Циклические и последовательные секции (разделы)

Иерархия программ разделена на четыре основные секции:

Таблица 1.1

Секция	Содержимое
BEGIN (начальная)	Программы, исполняемые в начале каждого цикла системы исполнения.
SEQUENTIAL (последовательная)	Программы, подчиняющиеся динамическим правилам SFC.
END (конечная)	Программы, исполняемые в конце каждого цикла системы исполнения.
FUNCTIONS (функции)	Набор подпрограмм, предназначенных для использования в любой программе любой секции проекта.

Программы начальной и конечной секций описывают циклические операции и не зависят от времени. Программы последовательной секции описывают последовательные операции, в которых таймерная переменная явно синхронизирует основные операции.

Главные программы начальной секции систематически выполняются в начале каждого цикла исполнения. Главные программы конечной секции систематически выполняются в конце каждого цикла исполнения. Главные программы последовательной секции выполняются в соответствии с динамическими правилами SFC.

Программы секции "функции" - это подпрограммы, которые могут вызываться любой программой проекта. Программа секции "функции" может вызывать другую программу из этой секции.

Главные и дочерние программы последовательной секции должны быть описаны на языке SFC. Программы циклических секций (начальной и конечной) не могут быть написаны на языке SFC. Любая программа в любой секции может иметь одну или более подпрограмм. Любая программа последовательной секции может иметь одну или более дочерних программ на языке SFC. Подпрограммы не могут быть описаны на языке SFC.

Программы начальной секции обычно используются для описания подготовительных операций на устройствах ввода для построения переменных высокого уровня, получаемых от устройства и передаваемых для обработки. Такие переменные часто используются программами последовательной секции.

Программы конечной секции обычно используются для операций защиты переменных, обработанных в последовательной секции перед посылкой значений в выходные устройства.

1.1.3 Язык описания

Программа может быть разработана на следующих языках:

- Графические языки:

- Язык последовательных функциональных схем (SFC) для операций высокого уровня.
- Язык функциональных блоковых диаграмм (FBD) для сложных циклических операций.
- Язык релейных диаграмм (LD) для булевых операций.

- Текстовые языки:

- Язык структурированного текста (ST) для любых циклических операций.
- Язык инструкций (IL) для операций низкого уровня.

1.1.4 Правила выполнения программ

ISaGRAF – это **синхронизированная** система. Переключение всех операций контролируется таймером. Период срабатывания таймера называется продолжительностью **цикла** (рис. 1.1).

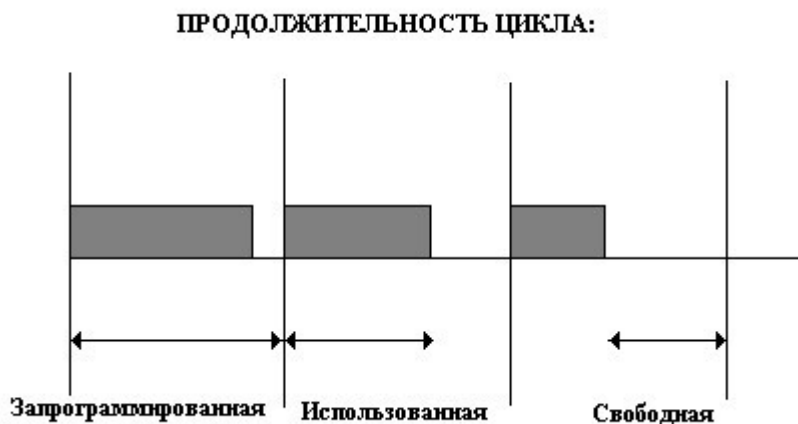


Рис. 1.1

Основными операциями, выполняемыми в течение одного цикла системы исполнения, являются следующие (рис. 1.2):

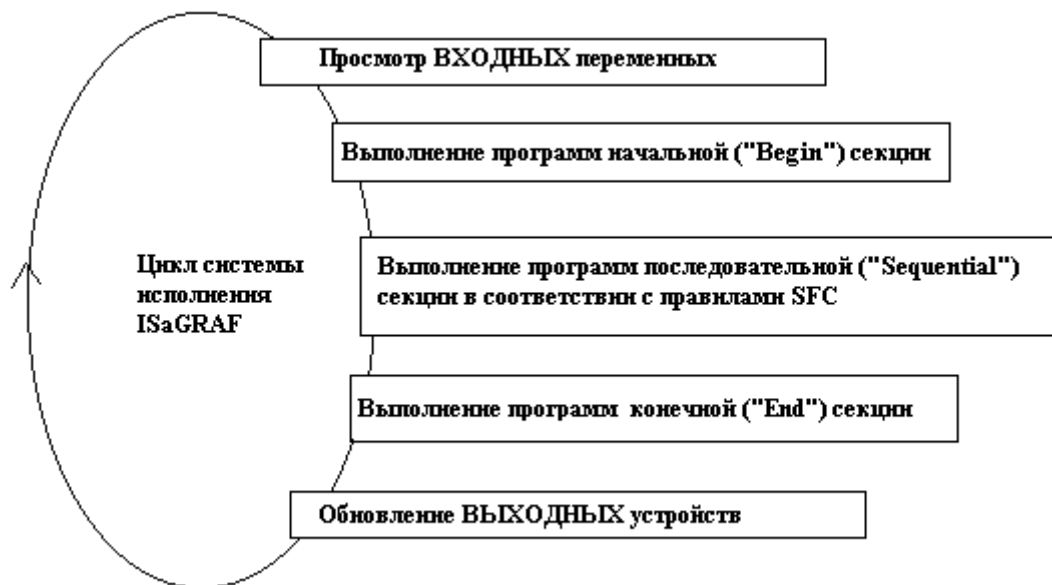


Рис. 1.2

Такая система позволяет:

- гарантировать, что в течение каждого цикла входная переменная имеет одно и то же значение;
- гарантировать, что обновление выходного устройства происходит в течение цикла только один раз;
- безопасно использовать разными программами одну и ту же глобальную переменную;
- вычислять и контролировать время реакции всей прикладной программы.

Основной принцип, который лежит в основе системы исполнения ISaGRAF Target – это синхронизация. Прикладная задача ISaGRAF (проект ISaGRAF) выполняется строго по временным циклам, продолжительность которых определяется при компиляции задачи.

Минимальная продолжительность циклов исполнения прикладной задачи определяется характеристиками аппаратно-программной платформы (ISaGRAF Target), на которой происходит исполнение задачи. Например, для OS-9 этот параметр равен 10 мс. Если же параметр синхронизации устанавливается равным нулю, то прикладная задача будет работать по принципу: выполнялась текущая программная последовательность - управление передается следующей без ожидания.

Принцип цикличности выполнения прикладной задачи реализуется следующим образом: в начале каждого временного цикла заданной продолжительности выполняются все программы/подпрограммы секции BEGIN, в конце цикла – все программы/подпрограммы секции END, а в середине цикла выделяется время для выполнения очередных шагов программ/подпрограмм секции SEQUENTIAL. На программы из секций BEGIN и END при необходимости возлагаются, как правило, задачи по обновлению входных и выходных переменных. Такая схема работы ISaGRAF-приложения гарантирует, что в рамках одного временного цикла будет работа только с одной копией входных и выходных переменных.

2 Задание

2.1 Инсталлировать инструментальную систему ISaGRAF. Для надежной и корректной инсталляции ISaGRAF следует избегать кириллических символов в имени пути (именах соответствующих папок) к инсталляционному файлу. Разместить в инсталлированной системе ISaGRAF демо-проект RFWASH путем копирования папки RFWASH в папку ISAWIN\SMP.

2.2 Открыть и ознакомиться с работой демо-проектов, включающих встроенную графику (анимацию): BOTTLEF, DEMO, RFSCOPE, SORTING, RFBARS, RFGRAPH, RFTMRFB, RFWASH.

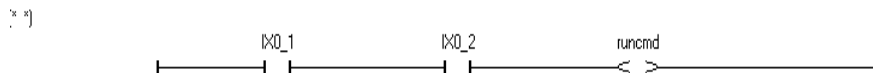
Приведем перечень демо-проектов с присоединенной графикой, запуск которых осуществляется двойным щелчком мыши по названию проекта и последующим за этим щелчком по кнопке 'Симуляция': BOTTLEF, DEMO, RFSCOPE, SORTING.

Приведем перечень демо-проектов с присоединенной графикой: RFBARS, RFGRAPH, RFTMRFB, RFWASH, для демонстрации графики которых требуется в дополнение к описанным выше действиям выполнить следующие: Инструменты\Прожектор\Файл\Открыть и в диалоговом окне присоединить указанный графический компонент.

На появившейся графической схеме для ее инициации возможно потребуется двойным щелчком мыши по соответствующим графическим элементам схемы установить логические переменные в истинное значение.

В частности, для проектов RFWASH, RFGRAPH на появившейся картинке нужно выполнить следующие действия: для проекта RFWASH сначала дважды щелкнуть мышью по надписи 'run' и в диалоговом окне установить логическую переменную в истинное значение, а затем при повторной инициализации графической схемы то же самое проделать с надписью 'soap'; для проекта RFGRAPH указанные действия проделать с надписью 'stop'. Для функционирования графической схемы проекта DEMO необходимо сначала заблокировать все входные переменные (графические элементы: общий рубильник, включатель и выключатель, левая и правая стрелка), а затем устанавливать соответствующие переменные в истинное значение. Для проекта RFTMRFB, демонстрирующего работу таймерных функциональных блоков, в выбранной графической схеме (TON, TOF или TP) следует сначала установить в окне симулятора с входными и выходными каналами соответствующий входной канал (TON Cmd, TOF Cmd или TP Cmd) для наблюдения переднего фронта сигнала, а затем сбросить его для наблюдения заднего фронта сигнала.

Программа теперь завершена. Вот полученный результат :



Выйдите из редактора и сохраните изменения при помощи команды «Выход» меню «Файл». Щёлкните «Да» для сохранения изменений.



Редактирование программы SFC

Для начала редактирования программы SFC «RunStop», дважды щёлкните мышью на её имени в окне Менеджера Программ или воспользуйтесь кнопкой «Редактировать программу».



Окно редактора SFC теперь открыто. Для увеличения рабочей области измените размер окна до полного экрана.



Начальный шаг уже существует и выбран. Нажмите стрелку "Вниз", чтобы выбрать пустую ячейку после начального шага (0,1)

F4 F3 Нажмите F4, затем F3, чтобы ввести переход и шаг.

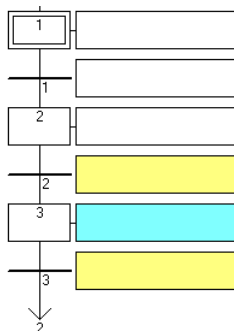
F4 F3 Нажмите F4, затем F3, чтобы ввести еще один переход и шаг.

F4 Нажмите F4, чтобы ввести еще один переход.

F5 Нажмите F5, чтобы ввести прыжок на шаг и выберите GS2 в качестве назначения прыжка.



Схема готова. Нажмите кнопку "Увеличение" на панели инструментов, чтобы увеличить размер ячеек и дать место для инструкций второго уровня. Вот схема:



Чтобы ввести программирование перехода 2, выберите его, используя стрелки клавиатуры, и нажмите клавишу "Enter". Программное окно уровня 2 открыто. Введите программу уровня 2 для перехода 2:

RunCmd;

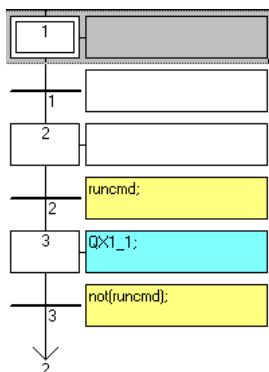
^TAB Нажмите клавиши "Control + Tab" чтобы передвинуть фокус назад на схему SFC (или закройте окно уровня 2) , сдвиньте выбор на шаг 3, и нажмите клавишу "Enter" чтобы редактировать текст уровня 2:

QX1_1;

И сделайте то же самое, чтобы ввести текст перехода 3:

Not (RunCmd);

^F4 Нажмите клавиши "Control + F4" чтобы закрыть окно уровня 2. Программа SFC теперь завершена.



Выйдите из редактора и сохраните изменения при помощи команды «Выход» меню «Файл». Щёлкните «Да» для сохранения изменений.



Создание кода приложения

Используйте меню «Создать» и команду «Создать приложение» из окна Менеджера Программ для создания кода (или кнопку "Создать код приложения" панели инструментов)

Когда генерация кода завершена, появится окно диалога, которое будет содержать вопрос: закончить генерацию кода *сейчас* (now) или *продолжить работу* (continue) : Нажмите кнопку «Выход».

Генерация кода (перекомпиляция) необходима после любых изменений в прикладной программе для фиксации этих изменений.



Симуляция

Используйте меню «Отладка» и команду «Симуляция» из окна Менеджера Программ для запуска ядра симулятора ISaGRAF (то же возможно при помощи соответствующей кнопки панели инструментов).

При появлении окна симулятора, приложение может быть протестировано (рис. 1.3). В данном примере входы 1 и 2 (зелёные кнопки – входные переменные IX0_1, IX0_2) должны быть установлены (значение – true) с помощью мыши оба (ярко-зеленый цвет) для запуска процесса (на выходе красный светодиод 1 загорается – выходная переменная QX1_1 при активности шага 3 получает значение true). Для удобства и лучшей визуализации отладки следует в пункте меню "Опции" окна симулятора с входными и выходными каналами установить соответствующие флажки (см. рис. 1.3).

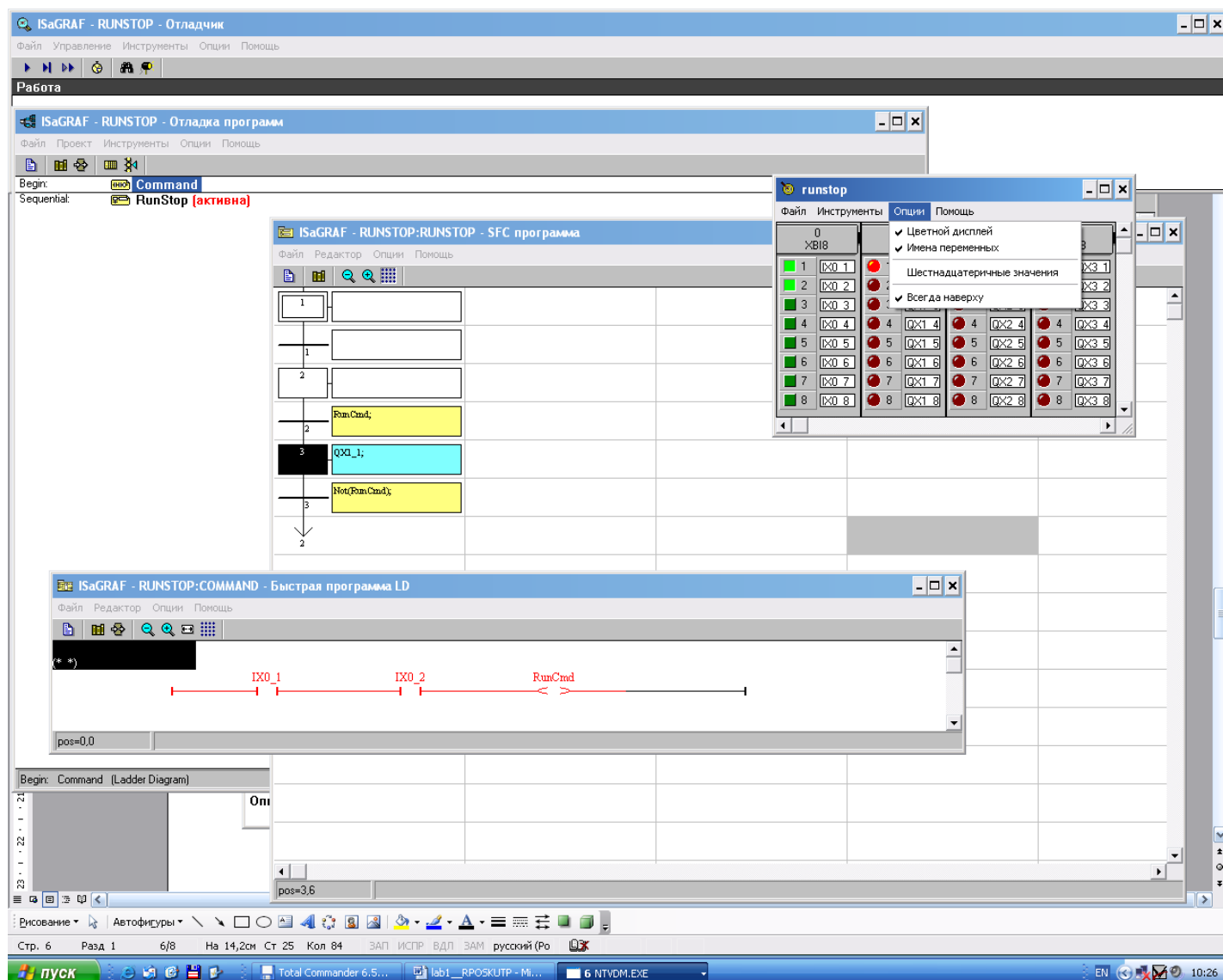


Рис. 1.3

Для дальнейшего продвижения маркера активности по схеме нужно сбросить хотя бы один из указанных входов (рис. 1.4).

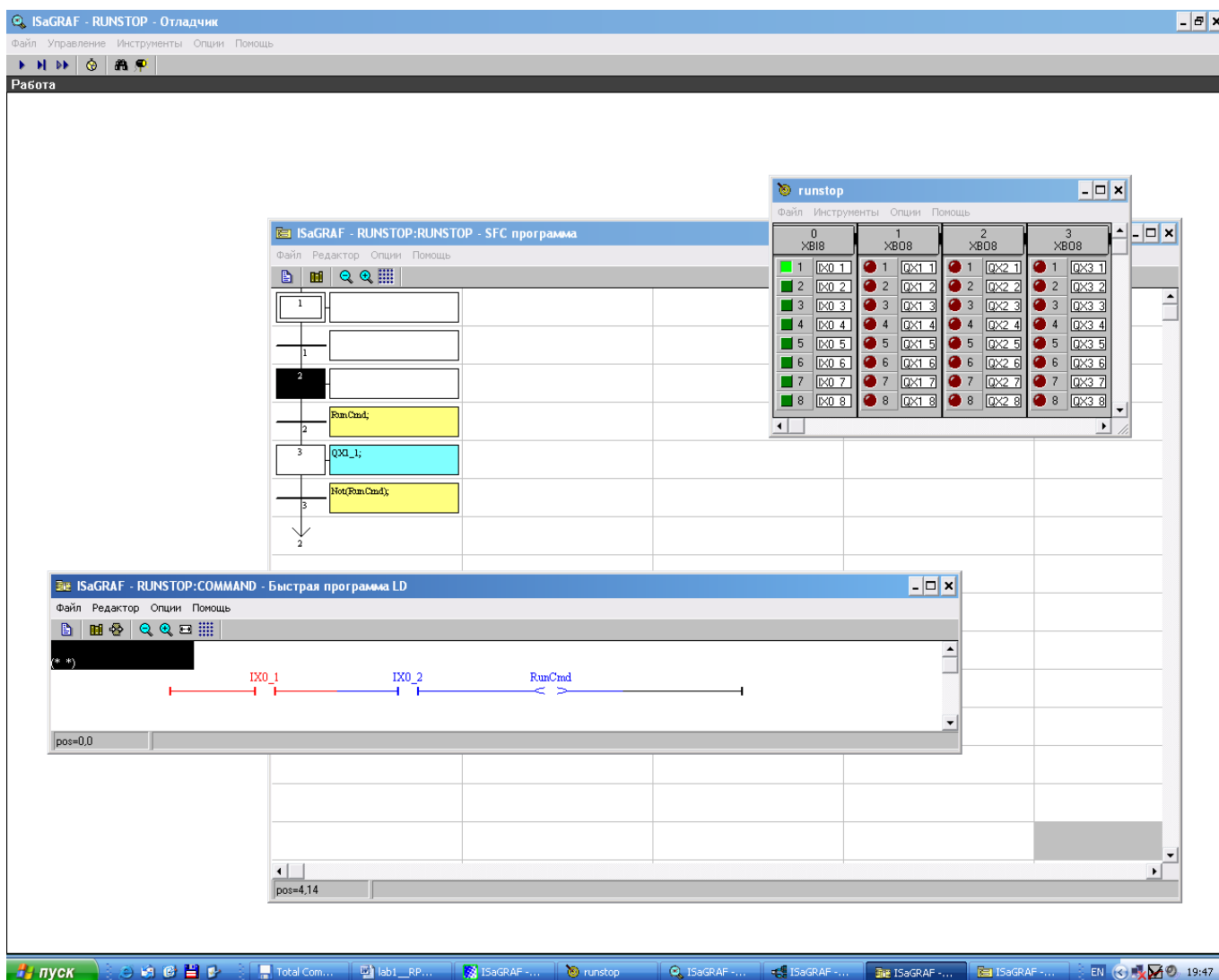


Рис. 1.4

После тестирования проекта в режиме реального времени следует протестировать его в пошаговом режиме. Для этого в окне отладчика с помощью кнопки "Пошаговый режим" установить данный режим, а затем использовать кнопку "Выполнить один цикл".

Закройте окно отладчика для выхода из симулятора: (команда «Файл» \ «Выход» или кнопка закрытия окна) либо закройте окно симулятора с входными и выходными каналами. В этом случае остальные открытые окна закрываются автоматически.

2.4 Создать графическую схему для проекта RunStop с использованием прожектора ISaGRAF.

Для удобства создания графической схемы желательно предварительно скопировать все необходимые графические элементы из папки lab1 graphic в папку проекта ISAWIN\SMP\RunStop.

Запустить проект RunStop на симуляцию. Затем выполнить команду Инструменты\Прожектор. Затем – команду Вставить\Картинка и выбрать для вставки фоновый рисунок background.bmp, подготовленный в редакторе Paint и расположенный в папке lab1 graphic или в папке RunStop (после копирования) (рис. 1.5). Для устранения эффекта запаздывания при отладке проекта по графической схеме объем файла фонового рисунка (файл с расширением bmp или wmf) по возможности следует ограничивать (желательно в пределах 1 Мб).

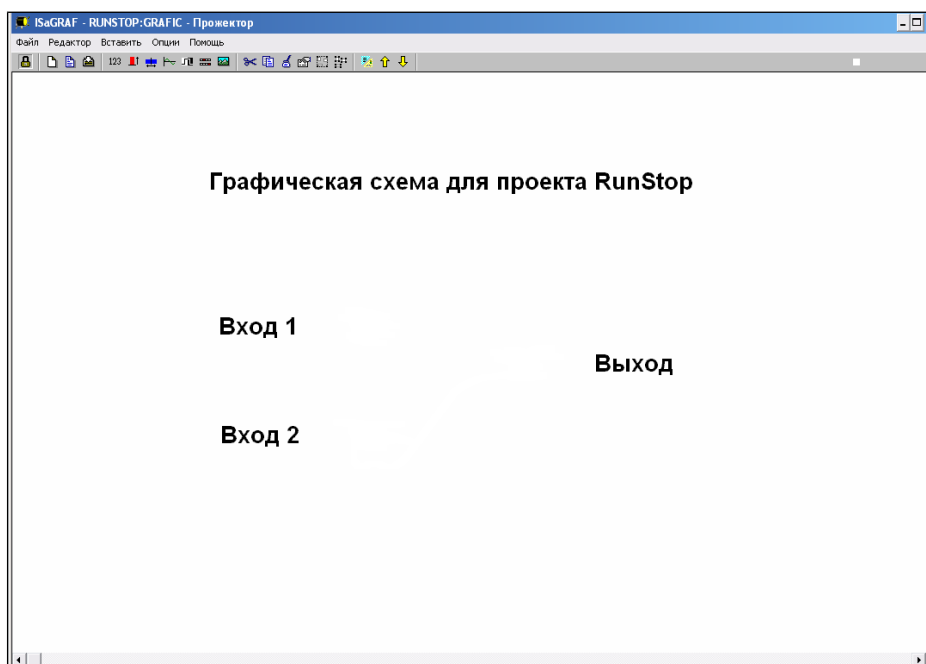


Рис. 1.5

Затем на данный фоновый рисунок следует наложить графические элементы для входных и выходных переменных проекта.

Сначала настроим графический элемент для входа 1. Для этого нужно щелкнуть мышью по кнопке "Булевская иконка" и в появившемся диалоговом окне в поле "Имя" с помощью щелчка мыши по кнопке "..." выбрать переменную IX0_1. Аналогичным образом в поле "False" выбрать в той же папке, что и фоновый рисунок, графический файл green0.ico, а в поле "True" – файл green1.ico. Установить флажок "Командная переменная". Щелкнуть по кнопке "Принять" и отбуксировать появившийся рисунок к надписи фонового рисунка "Вход 1" (правее) (рис. 1.6).

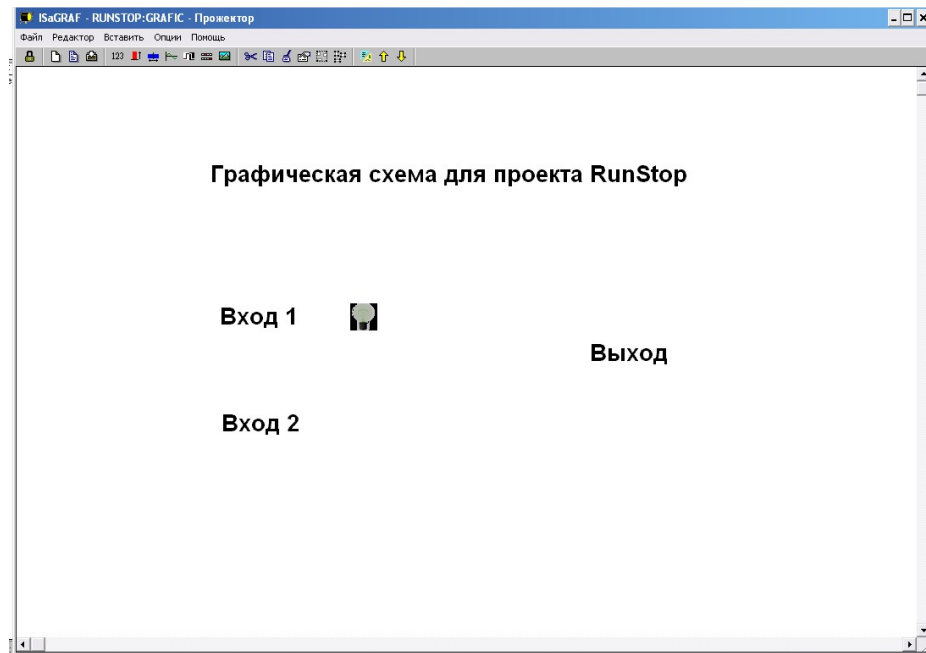


Рис. 1.6

Аналогичные действия следует проделать для надписи "Вход 2" с переменной IX0_2 и теми же графическими файлами green0.ico и green1.ico, а для надписи "Выход" - с переменной QX1_1 и графическими файлами red0.ico и red1.ico.

В итоге получается следующая графическая схема (рис. 1.7), которую нужно сохранить (Файл\Сохранить как..) под выбранным именем, закрыть окно симулятора и перекомпилировать проект.

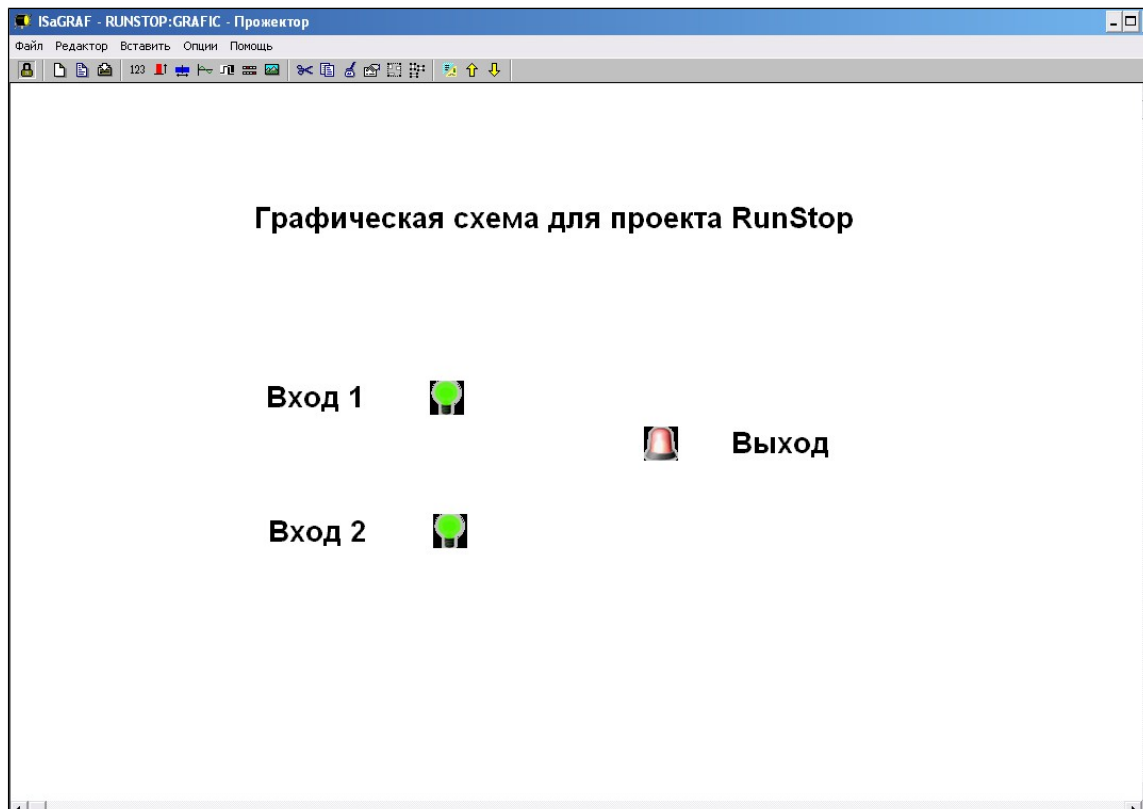
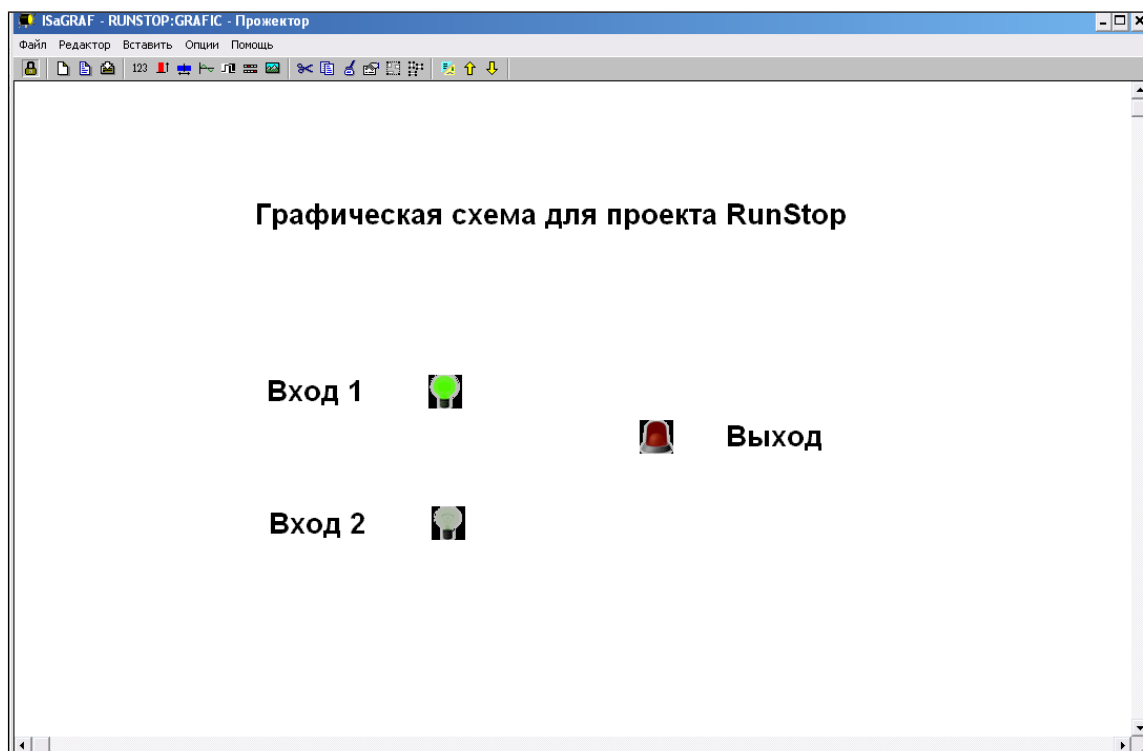


Рис. 1.7

Для удобства отладки проекта по графической схеме следует с помощью команды Отладка \ Рабочее пространство (окно "Программы") определить список документов для начального рабочего пространства. Такими документами могут быть программы, графика Прожектора, списки переменных. Документы, определенные в начальном рабочем пространстве автоматически открываются при запуске симуляции или отладки.

В появившемся диалоговом окне (рис. 1.8) следует с помощью кнопки ">>" переместить программы и графику из левого списка в правый и щелкнуть по кнопке "Принять".

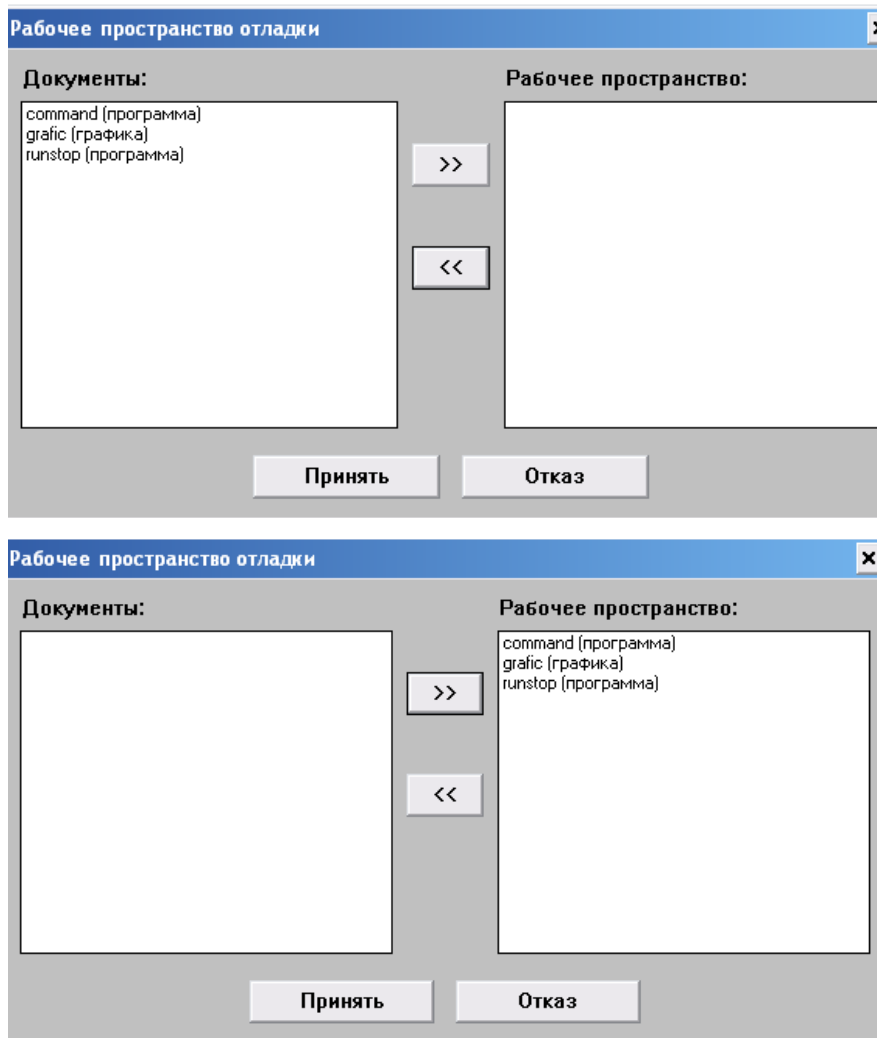


Рис. 1.8

Далее нужно запустить проект на симуляцию и с помощью кнопки "Блокировать/разблокировать" панели инструментов прожектора заблокировать графическую схему, затем заблокировать входные переменные (как в демо-проекте DEMO). После этого графическая схема готова к работе.

Далее следует свернуть окно симулятора (зеленые и красные индикаторы входных и выходных каналов) в панель задач и протестировать проект RunStop исключительно с использованием созданной графической схемы (рис. 1.7).

2.5 Ознакомиться с возможностями печати (Генератор документов) в среде ISaGRAF (Управление проектами\Проект\Печать – рис. 1.9). Для корректного отображения кириллицы следует установить для шрифта текста (формат страницы) и шрифта заголовка (заголовок страницы) шрифт Times New Roman или установить и воспользоваться шрифтами из папки Fonts. Также следует установить формат страницы, как это указано на рис. 1.4. В заголовке страницы можно указать в соответствующих графах штампа информацию о лабораторной работе, группе и исполнителях, а также можно указать путь к графическому файлу (черно-белый монохромный рисунок bmp), который заменит стандартный логотип CJ International (рис. 1.10). Для получения оптималь-

ного объема документа следует путем вырезания оставить только самое необходимое в составе документа: дескриптор проекта, иерархия программ, определения, переменные, исходный код (рис. 1.9).

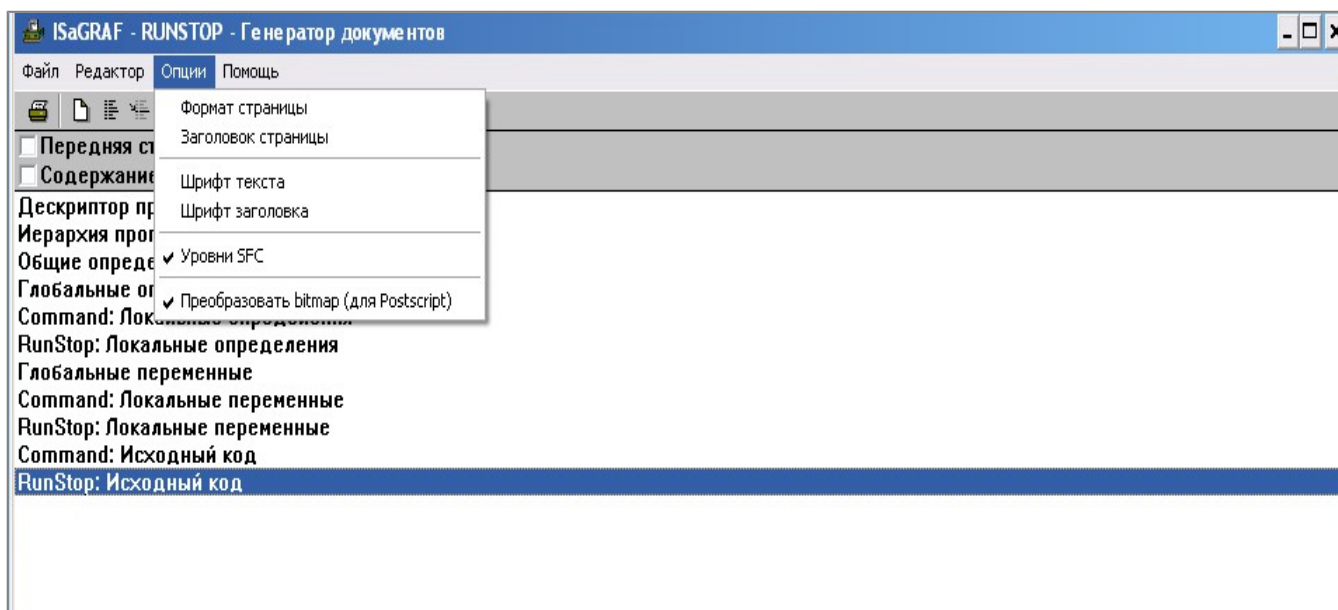


Рис. 1.9

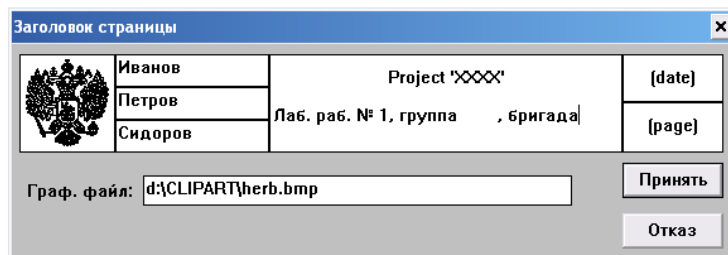
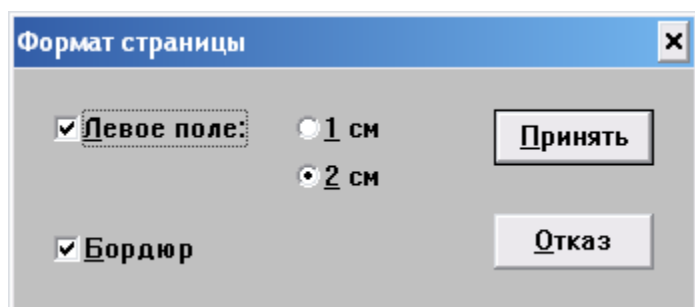


Рис. 1.10

Целесообразно при наличии пакета Microsoft Office в составе установленного программного обеспечения компьютера воспользоваться виртуальным принтером "Microsoft Office Document Image Writer" (Панель управления\Принтеры и факсы), для чего сделать его принтером по умолчанию.. После этого посланный на печать документ будет направляться в среду графического редактора Microsoft Office Document Imaging и сохраняться в виде файла с расширением mdi . Из среды указанного редактора после предварительного просмотра и возможной коррекции документ можно направлять на итоговую печать на реальный аппаратный принтер.

При наличии виртуального принтера "Adobe PDF" можно аналогично воспользоваться и им. В этом случае документ будет направляться в среду редактора Adobe Reader и сохраняться в виде файла с расширением pdf. Для корректного отображения кириллицы следует установить для шрифта текста (формат страницы) и шрифта заголовка (заголовок страницы) шрифт Calibri или Times New Roman.

Для перенесения документа в среду MS Word можно переносить листы документа по одному, выделяя и копируя каждый лист как изображение. Для перенесения всего документа сразу следует воспользоваться подходящим графическим редактором, например IrfanView как свободно распространяемым программным продуктом. Сначала нужно в среде редактора Microsoft Office Document Imaging документ в формате mdi сохранить в формате графического файла tiff, после чего открыть его в среде IrfanView. Далее следует использовать команду Options\Extract all frames... и, указав путь к папке назначения (Destination directory) и формат представления (Save as: - лучше jpeg), щелкнуть по кнопке Start. После этого останется в среде редактора MS Word осуществить вставку рисунка из файла с выделением всех страниц документа (графических файлов в папке назначения).

3 Содержание отчёта

- 1) Задание.
- 2) Распечатка с помощью генератора документов ISaGRAF отлаженного проекта RunStop, включая общую структуру проекта, отдельные его секции со всеми уровнями, список переменных, а также распечатка (скриншоты) итогов тестирования без графической схемы и с графической схемой.
- 3) Выводы по выполненной работе.