

Министерство общего и профессионального образования
Российской Федерации

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И
РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра электронных приборов (ЭП)

К ЗАЩИТЕ ДОПУСТИТЬ
Заведующий кафедрой ЭП
д-р физ.-мат. наук, проф.
_____ С.М. Шандаров

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ ПЕЧАТНЫХ ПЛАТ

Дипломная работа
ФЭТ ДР.3.532.001 ПЗ

СОГЛАСОВАНО

Консультант по экономике

_____ 1998 г.
«__» _____

Консультант по безопасности
жизнедеятельности
доцент каф. ЭП

_____ Мотошкин В.В.
«__» _____ 1998 г.

Студент гр. 353

_____ Сараев К.В.
«__» _____ 1998 г.

Руководитель
дипломной работы
главный инженер

ПО «Контур»
_____ Лоос М.А.
«__» _____ 1998 г.

Реферат

Дипломная работа 80 с., 11 рис., 12 табл., 8 источников, 5 приложений, 5 л. графического материала.

ПЕЧАТНЫЕ ПЛАТЫ, САПР, ЧПУ, МИКРОКОНТРОЛЛЕРЫ, ПЕРФОЛЕНТА.

Объектом исследования являются системы проектирования печатных плат, сверлильные станки с числовым программным управлением, устройства сопряжения, автоматизированные системы управления производственными процессами.

Цель работы - построение технологии автоматизированного проектирования печатных плат, сопряжение этой технологии с существующим технологическим оборудованием (сверлильными станками с ЧПУ).

В процессе работы производились теоретические и экспериментальные исследования отдельных составляющих и общей структуры.

В результате исследования была разработана концепция технологии автоматического проектирования печатных плат, создано устройство по сопряжению персональной ЭВМ и сверлильного станка с ЧПУ СМ-600. Написаны вспомогательные программы для сопряжения персонального компьютера типа IBM PC с фотоплоттером типа ЭММА 30 из комплекса QWEST, создано устройство для подключения перфоввода типа 'Консул' к компьютеру и разработано программное обеспечение для ввода и архивирования информации с перфолент. Решена проблема преобразования информации от программного комплекса КЛАРА в стандартный формат EXXELON.

Дипломная работа выполнена в среде разработки Turbo Pascal, Assembler MCS51, MultiEdit и оформлена в текстовом редакторе Microsoft Word 7.0.

Abstract

Degree work 80 pages, 11 pictures, 12 tables, 8 sources, 5 application, 5 pages of graphical material.

PRINTED CIRCUIT BOARD, CAD, CNC, MICROCONTROLLERS

Research object is CAD of PCB, machine-tool with numeric control system, interface board, CAD of industrial process.

Goal of work – building of CAD technology PCB, interfacing this technology with existence industrial NC machine.

During work were made theoretical and experimental research separate making and common structure.

As the research result were created concept of PCB CAD technology, were created device to interface personal computer with drilling machine CM-600. Wrote auxiliary program for interface personal computer IBM PC with plotter EMMA 30, for input and archived information from device 'Konsul'. Derived problem of information converting from program system 'Klara' to standard format EXCELLON.

The degree work was made in system of development of the programs Turbo Pascal and Assembler, MultiEdit, designed with Microsoft Word 7.0 text editor.

4. Содержание пояснительной записки (перечень подлежащих разработке вопросов)

5. Перечень графического материала (с точным указанием обязательных чертежей):

6. Консультанты по проекту (с указанием относящихся к ним разделов проекта):

7. Дата выдачи задания: _____

Руководитель _____
(подпись)

Задание принял к исполнению

(подпись студента)

Содержание

1 Введение	7
2 Концепция построения системы	9
2.1 Существующая технология ППП	9
2.2 Основные направления модернизации ППП	9
3 Взаимодействие с технологическим оборудованием	11
3.1 Введение	7
3.2 Разработка комплекса связи с ЧПУ	7
3.2.1 Возможности ЭВМ	12
3.2.2 Выбор микроконтроллера	13
3.2.3 Схема сопряжения	14
3.2.4 Управляющая программа для ведущей ЭВМ	15
3.2.5 Программа микроконтроллера схемы сопряжения	18
3.3 Подключение перфосчитывателя к ЭВМ	19
3.3.1 Схема связи ЭВМ и перфосчитывателя	19
3.3.2 Управляющая программа ЭВМ	20
4 Вспомогательные программы	21
5 Разработка организационно-экономических вопросов	22
5.1 Техничко-экономическое обоснование целесообразности исследования	22
5.2 Планирование комплекса работ. Сетевая модель	23
5.2.1 Назначение сетевой модели	23
5.2.2 Сетевая модель	23
5.2.3 Расчет параметров сетевого графика	27
5.2.4 Карта загрузки исполнителей	32
5.2.5 Расчет затрат на разработку проекта	35
6 Вопросы охраны труда	39
6.1 Анализ объективных факторов производственной опасности	39
6.2 Требования безопасности	45

6.2.2 Общие эргономические требования.....	46
6.3 Разработка мер безопасности на рабочем месте оператора ЭВМ...	49
6.4.1 Общие положения	49
6.4.2 Обязанности персонала при аварийных ситуациях.....	51
7 Заключение.....	52
8 Список использованных источников	53
Приложение А Текст программы MAINDRL.PAS	54
Приложение Б Текст программы DRILL51.ASM	61
Приложение В Текст программы PERFOVW.PAS	70
Приложение Г Текст программы C60TODRL.PAS	73
Приложение Д Принципиальная схема устройства сопряжения.....	80

Графический материал:

ФЭТ ДП 353000.001ЭЗ Устройство сопряжения. Схема электрическая принципиальная.	На отдель ных лис-
ФЭТ ДП 353000.002ПЗ Схема технологии проектироваия ПП	тах
ФЭТ ДП 353000.003ПЗ Комплекс СМ-600 до модернизации	
ФЭТ ДП 353000.005ПЗ Модернизированный комплекс СМ-600	
ФЭТ ДП 353000.004ПЗ Сетевой график. Карта загрузки исполнителей	

1. Введение

Электронная промышленность большинства развитых стран использует для производства продукции самые перспективные и совершенные технологии. Одной из составляющих производства такого рода является технология производства печатных плат для монтажа электронных устройств.

Проектирование рисунка проводников печатных плат совершенствовалось одновременно с развитием ЭВМ. Один из первых методов изготовления рисунка проводников - фотографическая пересъёмка с оригинала большего масштаба изготовленного вручную. С тех пор техника шагнула далеко вперёд. Однако технология производства в нашей стране заметно отстаёт от развитых стран.

В настоящее время для решения задач автоматизации проектирования печатных плат (ППП) в нашей стране повсеместно используются электронная вычислительная техника. Наиболее распространены комплексы на базе ЭВМ типа СМ и ЕС. Эти комплексы представляют из себя комплект из центральной ЭВМ и некоторого количества терминалов. При этом все вычисления производятся на головной ЭВМ, а терминалы выполняют роль устройств взаимодействия с пользователем. В настоящее время такая концепция в области систем автоматизированного проектирования себя не оправдывает. Это обуславливается всё возрастающими вычислительными потребностями отдельных задач САПР. Также пользователям значительно удобнее, когда их задачи решаются непосредственно местными ресурсами.

Одной из проблем с которой сталкивается руководитель предприятия с САПР на базе ЭВМ типа СМ и ЕС это дороговизна их обслуживания. Данная техника в основном работает на предприятиях уже значительное время. Поэтому частота поломок значительно возрастает по сравнению с начальным периодом эксплуатации. Для устранения поломок приходится содержать значительный штат

обслуживающего персонала. Специалистов в данной технике постепенно становится меньше, что является ещё одной трудностью для руководителей.

В связи с вышеизложенным понятно желание руководителей производств о переводе систем автоматизации ППП на более современные типы ЭВМ. При этом производится пересмотр всей концепции отделов САПР. Происходит отказ от идеи централизованной ЭВМ в сторону распределённой математической обработки.

При таком переходе возникает проблема выбора компьютерной платформы для создания требуемых математических мощностей. В современном мире ведущие позиции занимают IBM-совместимые компьютеры. У машин данного класса имеется значительная вычислительная мощность требуемая для решения задач САПР. Немаловажно и то, что для компьютеров такого типа существует значительная база разработанного программного обеспечения, наработан большой опыт работы с системами САПР на данной технике.

Однако при этом возникает множество проблем как организационного, так и технического характера. Так как спектр решаемых задач достаточно широк, то для их решения используется несколько программных комплексов. Поэтому возникает задача обмена результатами работы между разнородными программами .

Одна из технических проблем, это взаимодействие новых ЭВМ со старым периферийным оборудованием. В качестве последнего выступают различные станки с ЧПУ, автоматизированные производственные линии. Смена данных производственных комплексов на новые требует значительных стартовых финансовых затрат, что не многие современные отечественные производства могут себе позволить. Следовательно, возникает задача по связи современных ЭВМ с существующим технологическим оборудованием.

2. Концепция построения системы

2.1. Существующая технология ППП

Для понимания необходимых преобразований в системе проектирования ПП рассмотрим существующую технологию процесса.

Разработанная конструктором устройства принципиальная электрическая схема поступала разработчику печатной платы. После этого в течении длительного времени производилось трассирование рисунка печатной платы. Во время этого процесса возможно появление ошибок, которые сделают работу схемы невозможной. Затем нарисованная на миллиметровой бумаге топология печатной платы передавалась в отдел АСУ, где информация о топологии переводилась в электронную форму, т.е. происходила 'сколка' (на этом этапе также возможны ошибки). Полученная информация являлась достаточной для производства ПП. Электронный фотоплоттер выводит на фоточувствительную плёнку рисунок будущей ПП. Именно такая плёнка поступает в цех производства ПП. Однако для сверления отверстий в ПП используется станок под управлением ЧПУ. Входным носителем информации является перфолента.

2.1. Основные направления модернизации ППП

При переводе системы проектирования печатных плат с комплекса основанного на машине СМ-1420 под управлением ОС ДЕМОС и класса терминалов на комплекс состоящий из IBM-совместимых компьютеров возникли некоторые задачи без решения которых такой переход невозможен.

Для автоматизации проектирования ПП требуется выбрать программное обеспечение выполняющее такого рода функции. Программный комплекс PCAD позволяет выполнять автоматическую трассировку ПП, имеет постпроцессор для сопряжения с технологическим оборудованием, вводить электрические принципиальные схемы, конструктивы печатной платы. Выбор данного продукта обусловлено широким набором библиотек с отечественными электронными компонентами, наличием хорошо описанных форматов представления данных []. Практически, данный программный комплекс является стандартом 'de facto' в

электронной промышленности. Кроме того для автоматической трассировки соединений предполагается использовать программу 'ORCAD layout', которая кроме возможностей по вводу и размещению компонент (которые реализуются в системе PCAD) имеет достаточно гибкие и мощные средства трассировки ПП. При этом обеспечивается импорт из PCAD'а и экспорт обратно результатов разводки ПП. Однако, следует заметить, что для корректного выполнения данной операции требуется не добавлять информацию о ПП (соединения, элементы) в системе ORCAD Layout. Для полноты автоматизации ППП требуется использование ещё некоторых программных компонентов. Для окончательной проверки выводимого на фотоплёнку рисунка ПП предлагается использовать программу 'The CAM350 Family'. Данное решение позволяет выявлять ошибки на пути Проектирование ПП -> изготовление фотошаблона.

Другая задача при модернизации существующего комплекса это вывод управляющих перфолент для станков с ЧПУ. Ранее данная операция производилась путём выдачи данных на перфоратор входящий в комплекс СМ-1420. Но так как такой вид носителя информации в настоящее время является препятствием по внедрению новых технологий в САПР, то необходимо от него отказаться.

3. Взаимодействие с технологическим оборудованием

3.1. Введение

На производственном объединении "Контур" одним из потребителей данных на перфоленте является сверлильно-фрезеровальный станок с ЧПУ СМ-600. Данный станок представляет из себя комплекс состоящий из самого станка, который имеет 4 операционных узла; машины числового программного управления; устройства периферийного ввода УСЛ-300. На рисунке 2.1 приведён эскиз данного комплекса.

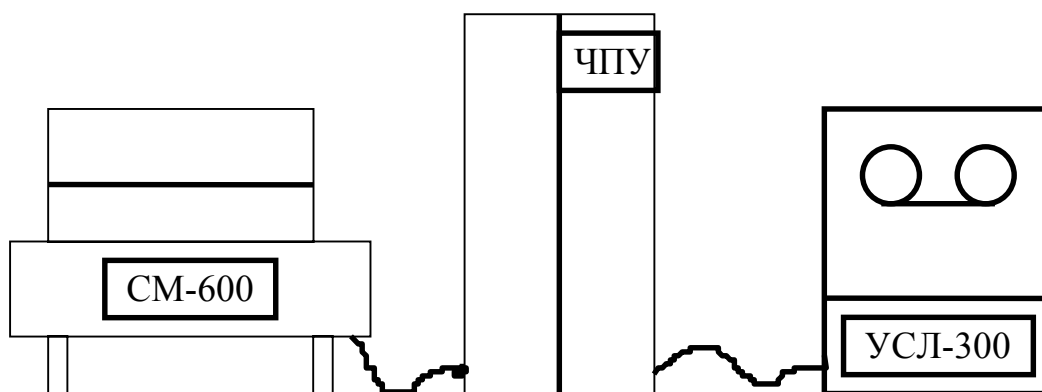


Рисунок 2.1 Комплекс СМ-600

Устройство УСЛ-300 представляет из себя комплекс из перфоратора, считывателя с перфоленты, управляющего контроллера. Очевидно, что для модернизации комплекса СМ-600 следует отказаться от использования устройства периферийного ввода УСЛ-300 и заменить его компьютером, который будет эмулировать его работу. При этом, в качестве входной информации должен выступать файл с данными по сверловке печатных плат. Данный файл поступает с системы проектирования печатных плат (САПР ПП) PCAD которая функционирует на платформе IBM-совместимых компьютеров.

На рисунке 2.2 представлен эскиз комплекса после модернизации.

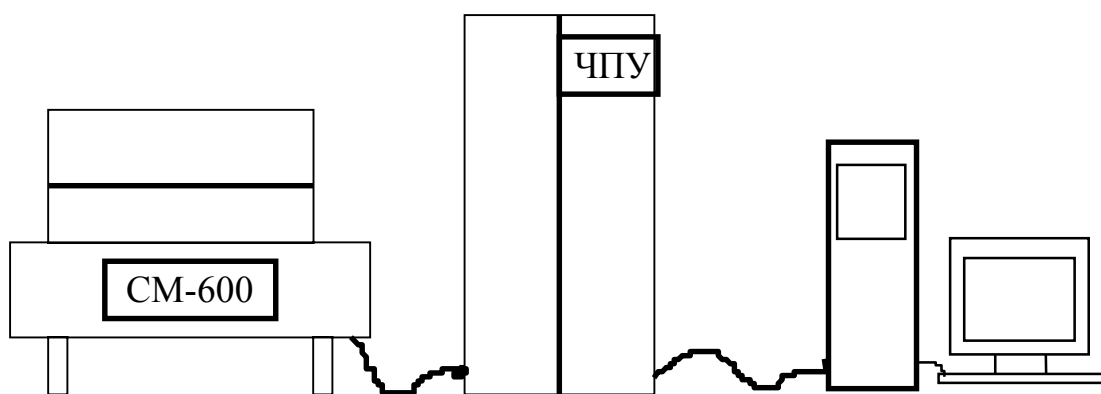


Рисунок 2.2 Комплекс СМ-600 после модернизации

3.2. Разработка комплекса связи с ЧПУ

3.2.1. Возможности ЭВМ

Для состыковки персональной ЭВМ и комплекса СМ-600 требуется объединить эти устройства посредством управляющей электрической связи. Предварительно был проведён анализ схемы электрической принципиальной устройства периферийного ввода УСЛ-300, которая входит в систему СМ-600 [8]. При этом выяснилось, что связь между ЧПУ и УСЛ-300 осуществляется путём использования 8-проводной шины данных и нескольких линий управления, которые позволяют вести контролируемый обмен данными между данными устройствами. По шине данных информация передаётся в одном направлении: от УСЛ-300 к ЧПУ, сигналы же управления имеют разную направленность.

Для решения проблемы связи необходимо рассмотреть аппаратные возможности персонального компьютера. Для снижения цены данного решения и повышения конкурентоспособности системы в целом необходимо максимально использовать базовые возможности ЭВМ. Одной из таких возможностей является параллельный интерфейс для связи с периферийными устройствами (принтерный порт CENTRONIX). Однако при проведении работ [] с использованием такого подхода было выяснено, что параллельный интерфейс ЭВМ имеет низкую нагрузочную способность. В условиях цеха это приводит к неработоспособности

устройства при воздействии внешних факторов (изменение напряжения питания, влажности...).

После проведения пробных испытаний было решено использовать последовательный порт RS232 ЭВМ и микроконтроллер, который принимает команды ЭВМ по этому интерфейсу и взаимодействует с ЧПУ. Эскиз построения этой системы приведён на рисунке 3.1.

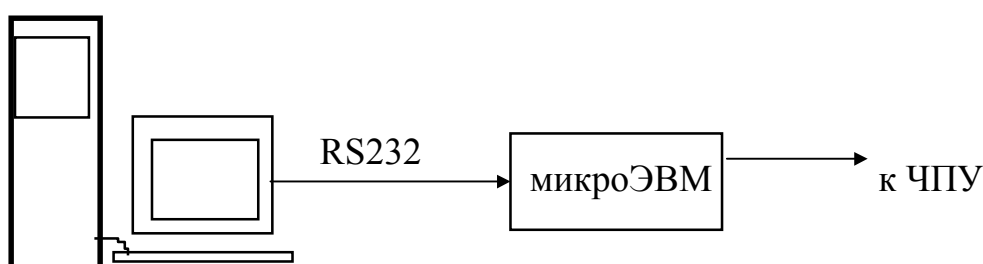


Рисунок 3.1 Интерфейс ЭВМ - ЧПУ

3.2.2. Выбор микроконтроллера

В настоящее время в мире наблюдается подъём в области разработки и производства периферийных микроконтроллеров (МК) . На рынке предлагается множество семейств такого рода устройств. Среди наиболее распространённых следует упомянуть 51-совместимые МК, PIC (фирма Microchip), 6805 (Motorola), Z8 (Zilog). В нашей стране получили наибольшее распространение МК совместимые с INTEL 8051. Во время обучения в институте нам был прочитан курс по программированию микроконтроллеров 8051. Многие фирмы в мире выпускают МК совместимые с 8051 (Интеграл, Philips, Dallas, Atmel...). После рассмотрения многих вариантов мною были выбраны МК 89c51 фирмы Atmel. Этот выбор обусловлен наличием электрически перепрограммируемой памяти программ (что значительно ускоряет отладку устройств), мощных

программируемых выводов (ток логического 0 до 20 ma), дешевизной и доступностью микросхемы.

Некоторые из возможностей МК 89с51:

1. 2 kb перепрограммируемой памяти программ
2. 32 программируемых вывода
3. Частота МК до 24 мегагерц
4. Наличие двух таймеров/счётчиков
5. Асинхронный последовательный порт совместимый с RS232

Именно наличие асинхронного порта стало последним доводом за МК 89с51.

3.2.3. Схема сопряжения

Микроконтроллер 89с51 имеет очень широкие возможности по управлению различными устройствами. Для реализации интерфейса МК - ЧПУ необходимо использовать программируемые линии ввода-вывода МК. Требуется 8 линий на вывод данных в ЧПУ, 2 линии на вывод - STROBE и READY, А также входные линии Пуск Цикла и Ввод.

Следует несколько более подробно рассмотреть протокол по которому происходит обмен между МК эмулирующим работу устройства УСЛ-300 и ЧПУ. Данный протокол основан на передаче по управляющим линиям сигналов готовности к приёму информации и сигналов достоверности информации на информационной шине данных в текущий момент. Для большего понимания протокола работы на рисунке 3.1 отображено направление передачи управляющих сигналов.

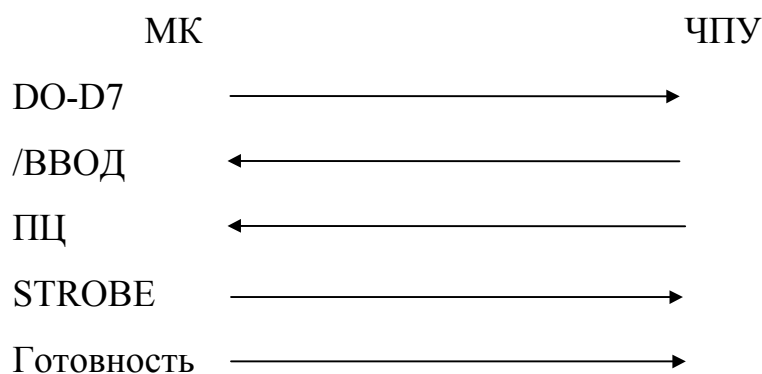


Рисунок 3.1 Направление передачи управляющих сигналов

На рисунке 3.2 представлена диаграмма работы протокола обмена по физическим линиям используемым в данных устройствах.

Именно по такому протоколу передаёт данные программа записанная в микроконтроллер. Однако для работы с последовательным портом ЭВМ требуются сигналы с амплитудой до 12 вольт, поэтому применён преобразователь уровней TTL-RS232 от фирмы Dallas - DS232A.

Принципиальная схема устройства сопряжения представлена в Приложении Е.

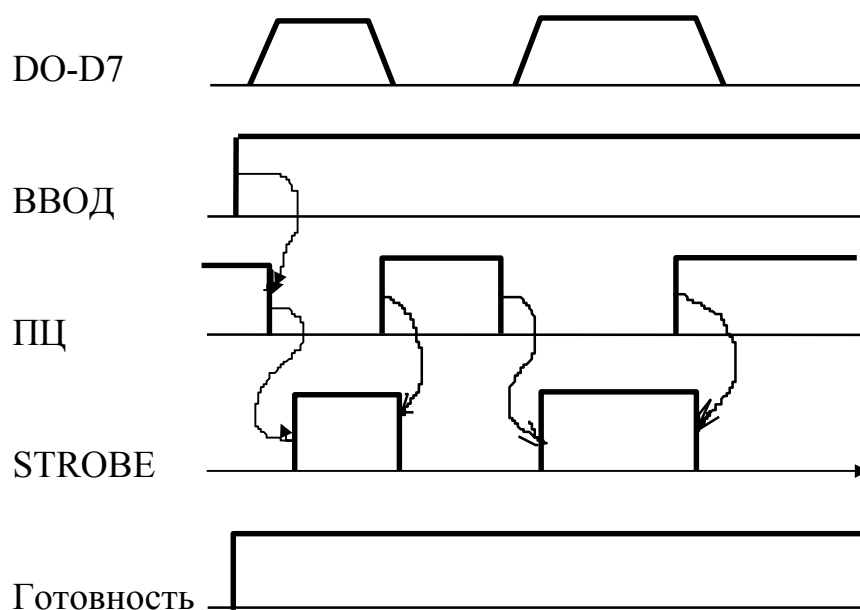


Рисунок 3.2 Диаграмма работы информационного протокола

3.2.4. Управляющая программа для ведущей ЭВМ

Ведущая ЭВМ в этом комплексе решает несколько возложенных на неё задач. Рассмотрим последовательно основные.

Так как инженерно-технический и цеховой комплекс завода разделены по нескольким зданиям, то возникает проблема переноса информации от её поставщика (конструктора) к исполнителю (цех производства ПП, комплекс СМ-600). Для решения этой проблемы предложено передавать требуемую информацию посредством компьютерных сетей. Для этого необходимо связать здания завода в единую информационную сеть. В данный момент часть этих работ уже проведена. Таким образом конструктор пересылает посредством электронной почты подготовленную информацию на ведущую ЭВМ комплекса СМ-600. Далее управляющая программа ЭВМ должна по запросу передать эту информацию посредством МК в ЧПУ. После чего начинается процесс сверловки по заданной программе отверстий в ПП.

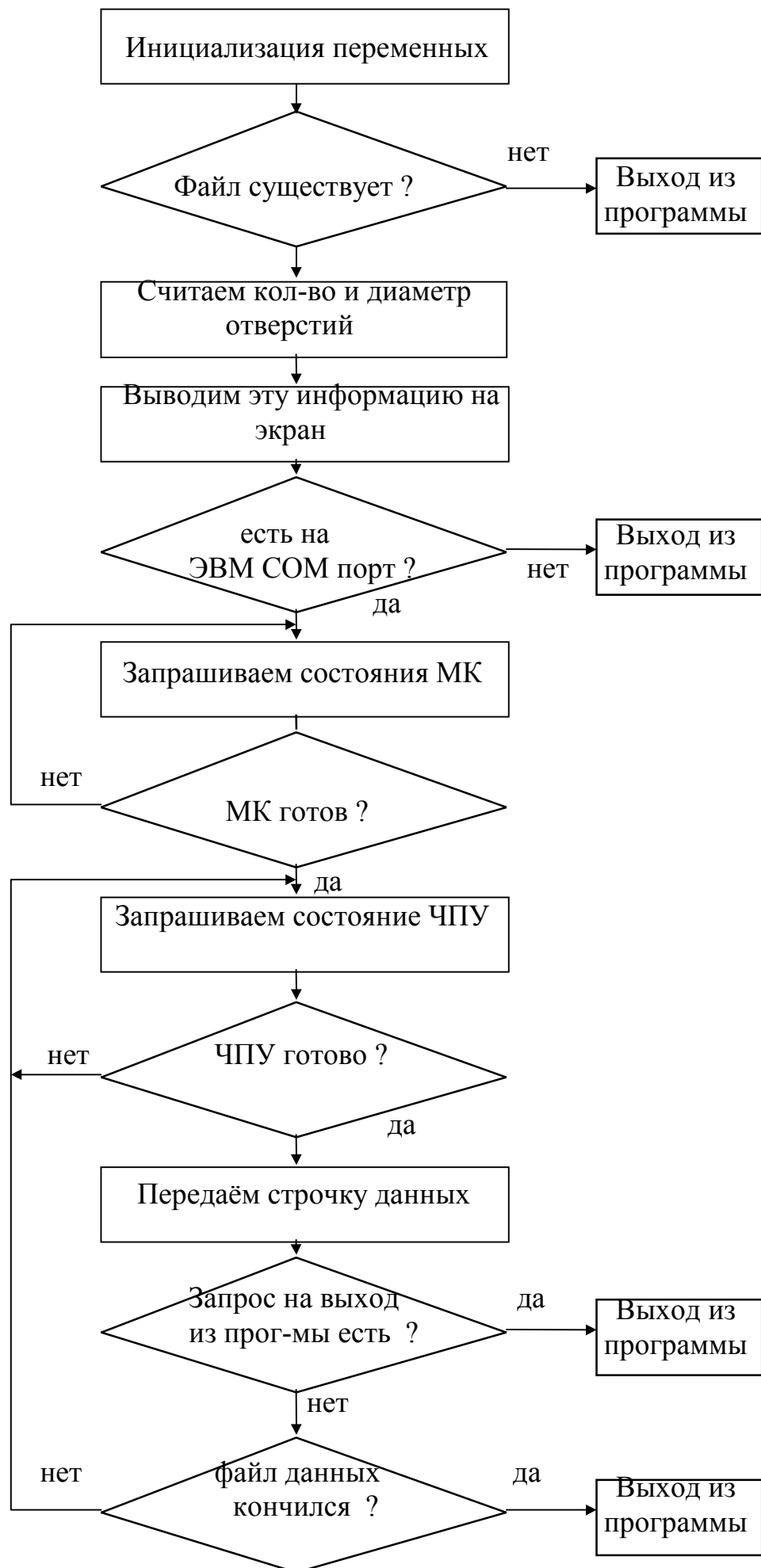
Управляющая программа ведущей ЭВМ должна обеспечивать необходимые сервисные функции, такие как: подсчёт количества отверстий каждого диаметра, прерывание передачи по требованию оператора, графическую индикацию количества переданной информации.

Для реализации всех этих возможностей была написана программа на языке TURBO PASCAL [1,2]. Выбор языка основан на его доступности для понимания и малых временных затратах на обучение. Программа обеспечивает взаимодействие с обслуживающим персоналом и управление передачей информации в ЧПУ. Алгоритм работы программы изображён на рисунке 3.3.

При написании программы использовалась информация по программированию последовательного порта персональной ЭВМ приведённая в литературе [3].

Одним из неожиданных эффектов при испытании программы стало резкое возрастание скорости передачи данных в ЧПУ. Это объясняется тем, что при работе с перфолентой скорость ограничивалась физическими возможностями по протягиванию перфоленты и прочностью самой бумаги из которой изготовлена перфолента. Но с заменой УСЛ-300 на ЭВМ и МК такое ограничений исчезло и скорость возросла с 300 байт в секунду до нескольких тысяч. Текст программы с подробными комментариями приведён в приложении А.

Рисунок 3.3
Алгоритм
управляющей
программы



3.2.5. Программа микроконтроллера схемы сопряжения

Программа записанная во внутреннюю память программ МК 89с51 написана на языке АССЕМБЛЕР так как для подобных микроЭВМ он является основным. Алгоритм должен обеспечивать передачу данных в ЧПУ согласно протоколу изображенному на рисунке 3.2, а также приём и исполнение команд по интерфейсу RS232 от управляющей ЭВМ. Текст программы DRILL51 приведён в приложении Б. На рисунке 3.4 представлен алгоритм работы программы.

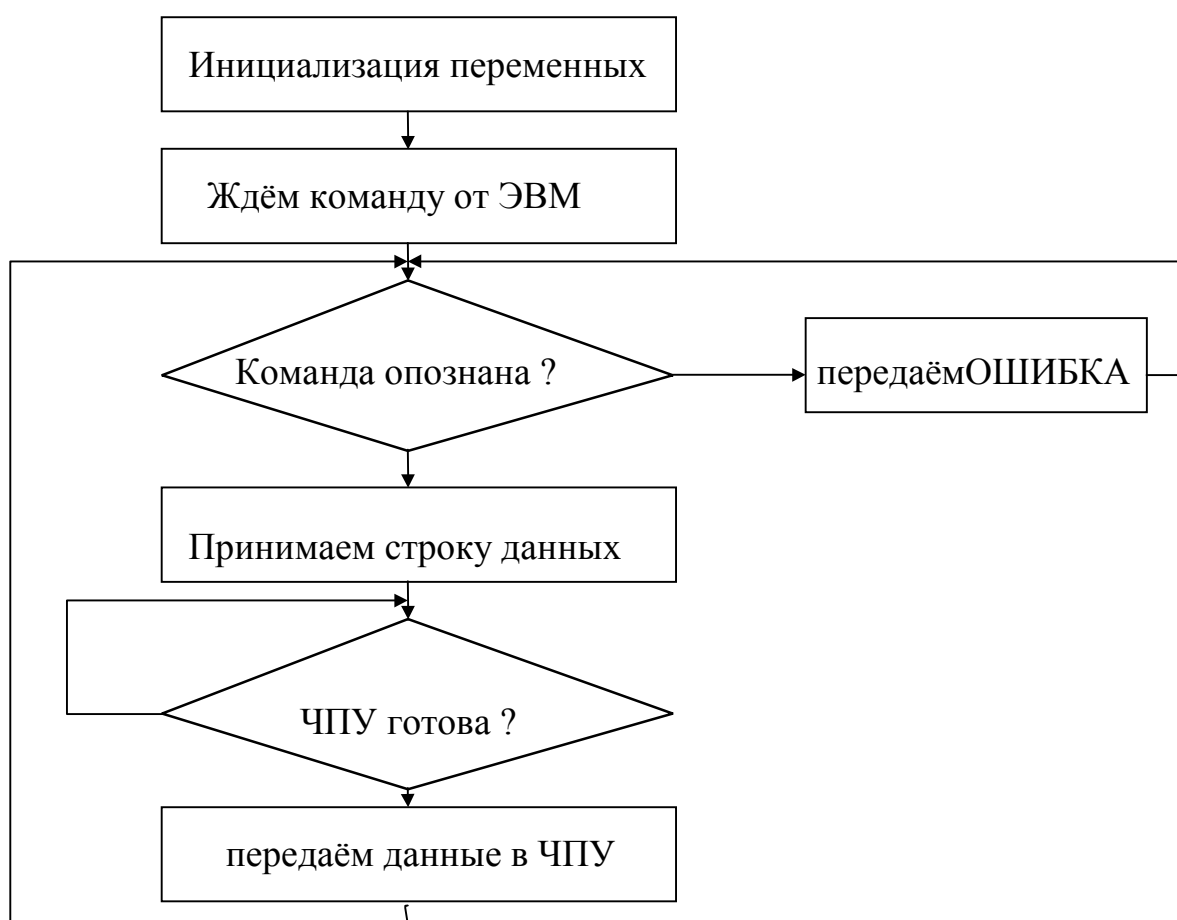


Рисунок 3.4 Алгоритм программы DRILL51 МК схемы сопряжения

3.3. Подключение перфосчитывателя к ЭВМ

3.3.1 Схема связи ЭВМ и перфосчитывателя

Несмотря на то, что после внедрения описанного выше комплекса необходимость в перфоленте, как носителя информации, отпала, необходимо предусмотреть возможность вводить с неё информацию в ЭВМ. Это обусловлено наличием большого количества информации накопленной на этом виде носителя. Наиболее рационально просто сохранять образ ленты в ЭВМ, не обращая внимание на то, какого рода информация на ней записана (программы для ЧПУ, данные для ПЗУ различных ЭВМ и т.д.). Таким образом получится универсальная программа не зависящая от считываемых данных. В качестве считывателя было подключено устройство считывания с перфолент 'КОНСУЛ'. Для связи с ЭВМ использовался простой переходник распайка которого приведена на рисунке 3.5.

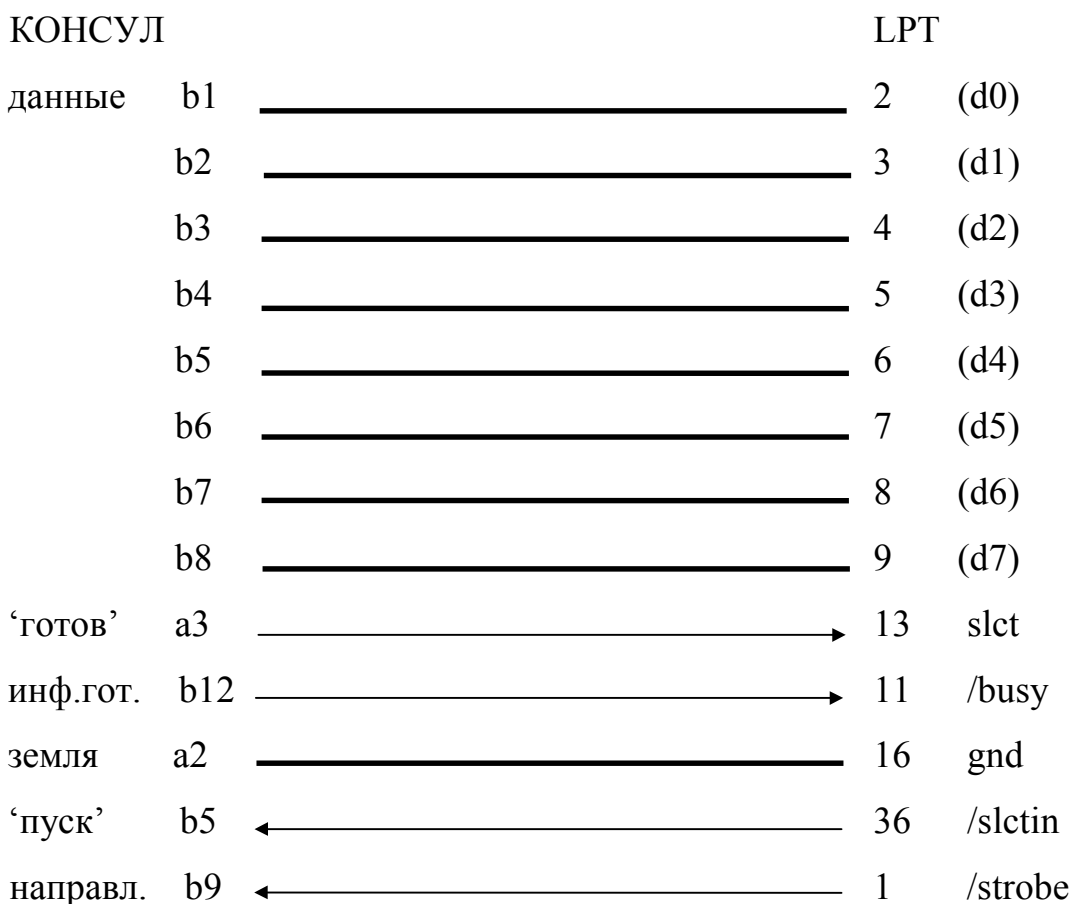


Рисунок 3.5 Схема распайки соединительного кабеля.

3.3.2 Управляющая программа ЭВМ

Необходимо отметить, что параллельный порт LPT является стандартным устройством в ЭВМ [] и присутствует на каждом IBM-совместимом компьютере. Соответственно отпадает необходимость в создании собственных устройств. При создании программы учитывалась необходимость запоминать комментарии к каждой считанной ленте для последующей каталогизации. Алгоритм программы представлен на рисунке 3.6. Текст программы PFLOAD написанной на языке Turbo Pascal представлен в Приложении Д.

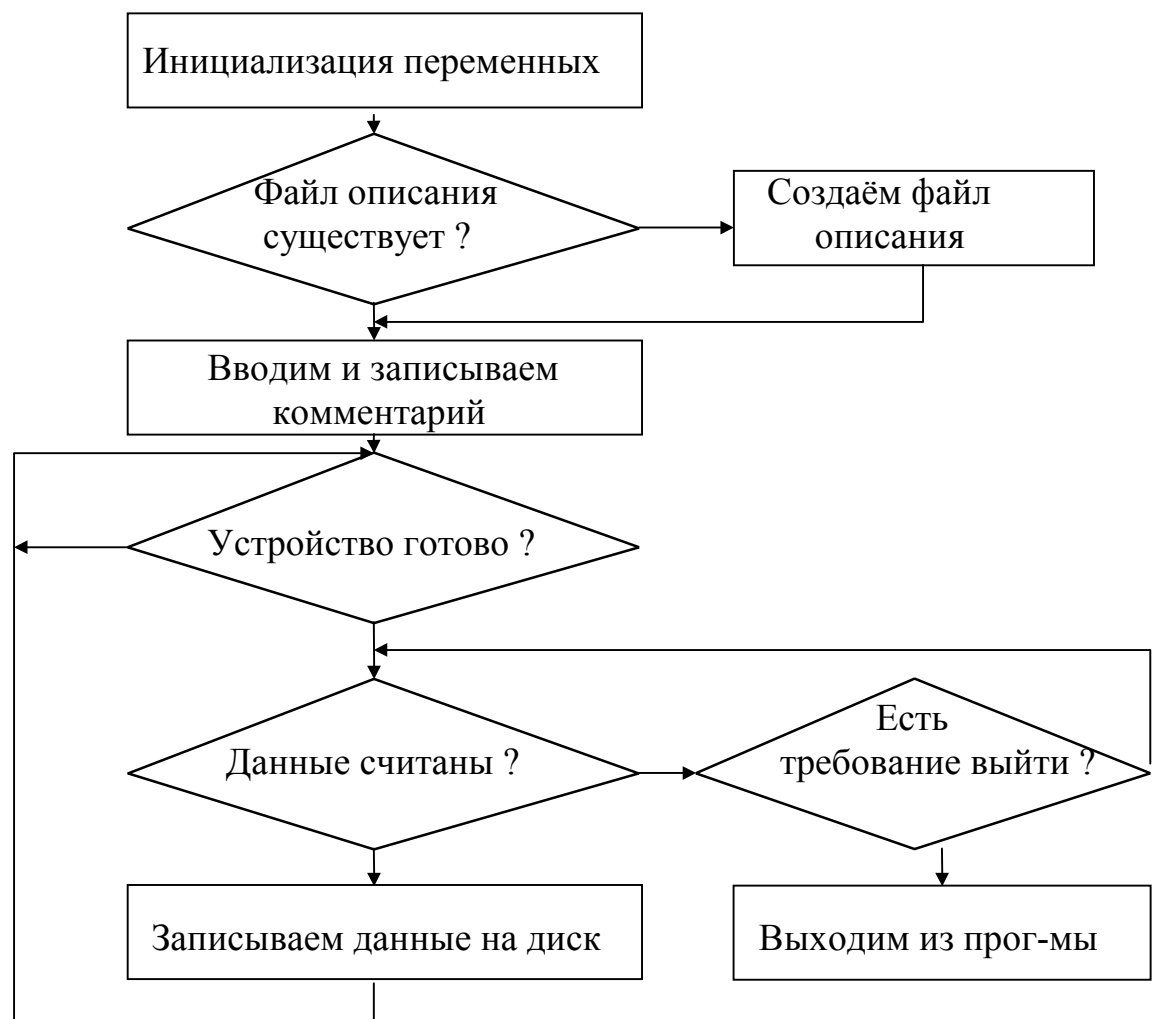


Рисунок 3.6 Алгоритм работы программы считывания данных с перфоленты

4. Вспомогательные программы

В процессе написания основной управляющей программы возникла необходимость просмотра файла образа перфоленты на экране компьютера. Это было достигнуто путём написания программы, которая формирует изображение перфоленты на экране в соответствии с заданным файлом образа перфоленты. Текст данной программы приведён в приложении В.

Необходимо заметить, что в некоторых подразделениях ПО 'Контур' используется технологическая программа КЛАРА разработанная отечественными специалистами. Она имеет свой собственный формат выходного файла с данными для сверления отверстий. Поэтому была написана программа перекодировщик C60TODRL которая позволяет переводить файл формата системы КЛАРА в файл формата программного комплекса PCAD. Текст данной программы написанной на языке Turbo Pascal приведён в Приложении Г.

5 Разработка организационно - экономических вопросов

5.1 Техничко - экономическое обоснование целесообразности исследования

Автоматизированне проектирование печатных плат широко применяется в настоящее время. Однако существует несколько различных способов для создания современной технологии проектирования. Некоторые программы входящие в комплекс проектирования ПП разработаны совсем недавно и возможность их интеграции в существующую технологиюещё недостаточно изучена. Изучение современного программного обеспечения может привести к созданию совершенно новой идеологии построения систем автоматизации проектирования печатных плат.

Кроме того, необходимо отметить, что в последнее время значительно возрос интерес к программным продуктам, которые позволяют провести весь процесс проектирования на ЭВМ, не прибегая к устройствам технологического вывода

Настоящая работа находится на стадии научно-исследовательских работ, о чем свидетельствует таблица 5.1.

Таблица 5.1 - Признаки

Признаки классификации		
Конечный результат	Теоретические выкладки, алгоритмы	+
	Лабораторный, опытные и макет. образцы	+
Предлагаемое внедрение		+
Сфера внедрения и использования	Производственная	+
	Непроизводственная	+

Наличие основы для других НИР		+
Наличие принципиально новых научных результатов		+
Метод исследования	Теоретический	+
	Экспериментальный	+
Источники финансирования	Средства компании «Стек», выделяемые на исследования	-
	Договорные	+

5.2 Планирование комплекса работ. Сетевая модель

5.2.1 Назначение сетевой модели

Система сетевого планирования и управления (СПУ) широко используется в промышленности строительстве, научных и проектных институтах. Она позволяет наглядно представить взаимосвязь работ, взаимозависимость между отдельными элементами системы.

СПУ - комплекс графических и расчетных методов, а также организационных мероприятий, которые обеспечивают моделирование, анализ и динамическое изменение планов выполнения проектов и разработок.

Сетевые модели относительно просты, удобны и позволяют быстро находить наилучшие варианты управляющих воздействий.

Основной носитель информации в сетевой модели - *сетевой график*, который наглядно представляет взаимосвязь отдельных работ между собой.

5.2.2 Сетевая модель

Сетевая модель, представляющая логическую взаимосвязь работ, изображается графически в виде сетевого графика, состоящего из стрелок, обозначающих работу, и кружков, обозначающих событие.

Работа - элемент сетевого графика, отражающий определенный четко выраженный этап производственного процесса, требующий затрат ресурсов и времени или только времени.

Событие - результат одной или совокупный результат нескольких работ, который дает возможность начать одну или несколько работ.

При построении сетевой модели каждую работу и событие всего комплекса работ сопоставляют элементам сети так, чтобы сохранилась их логическая последовательность.

Первый шаг при создании сетевого графика - составление перечня всех работ и событий, включенных в комплекс (см. табл.5.2, табл.5.3).

Таблица 5.2 - События комплекса работ

	Событие
1	Техническое задание (ТЗ) утверждено
2	Технико - экономическое обоснование исследований завершено
3	Изучение технологий проектирования печатных плат завершено
4	Функциональная схема программного продукта завершена
5	Компьютер подготовлен к установке программного обеспечения
6	Программное обеспечение установлено
7	Принципиальная схема устройства разработана
8	Исследования проведены
9	Результаты работы обработаны
10	Вопросы охраны труда разработаны

	Событие
11	Организационно-экономические вопросы разработаны
12	Отчет составлен
13	Отчет согласован
14	Отчет утвержден

Таблица 5.3 - Работы, входящие в комплекс

Код работы	Название работы
0-1	Утверждение технического задания
1-2	Технико-экономическое обоснование
2-3	Изучение технологий проектирования печатных плат
3-4	Разработка функциональной схемы программы
3-5	Подготовка компьютера к установке программного обеспечения
4-6	Настройка компьютера
5-6	Установка программного обеспечения
4-7	Разработка функциональной схемы исследований
6-7	Разработка принципиальной схемы устройства
7-8	Экспериментальные исследования
8-9	Обработка результатов экспериментов
9-10	Разработка вопросов охраны труда
9-12	Подготовка демонстрационных листов

Код работы	Название работы
10-11	Разработка организационно-экономических вопросов
11-12	Составление отчета
12-13	Согласование отчета
13-14	Утверждение отчета

Следующий этап построения сетевого графика - определение времени выполнения каждой работы. Реальная длительность работы зависит от множества факторов и имеет вероятностный характер, поэтому, обычно используют способы 2-х или 3-х оценок. Во втором случае используют следующие оценки:

- *оптимистическая оценка* - это минимальная продолжительность работы при наиболее благоприятном стечении обстоятельств t_{\min} ;
- *пессимистическая оценка* - максимальная продолжительность работы при наиболее неблагоприятном стечении обстоятельств t_{\max} ;
- *наиболее вероятная оценка* - продолжительность работы при нормальных, чаще всего встречающихся условиях $t_{\text{нв}}$.

Ожидаемая продолжительность выполнения работы определяется выражением:

$$t_{\text{ож}} = \frac{1}{6} (t_{\min} + 4 t_{\text{нв}} + t_{\max}) \quad (5.1).$$

Результаты расчетов представлены в таблице 5.4.

Таблица 5.4 - Результаты расчетов

Код работы	Продолжительность работы, дней			
	t_{\min}	t_{\max}	$t_{\text{нв}}$	$t_{\text{ож}}$
0-1	1	3	2	2
1-2	2	4	3	3
2-3	3	5	4	4
3-4	1	2	1	1
3-5	1	3	2	2
4-6	1	2	1	1
5-6	1	2	1	1
4-7	1	2	1	1
6-7	1	3	2	2
7-8	6	10	8	8
8-9	1	3	2	2
9-10	3	5	4	4
9-12	6	10	8	8
10-11	4	6	5	5
11-12	5	7	6	6
12-13	3	5	4	4
13-14	2	4	3	3

5.2.3 Расчет параметров сетевого графика

К основным параметрам сетевого графика относят критический путь, полный и свободный резервы времени выполнения работ.

Критический путь - наиболее протяженный путь, ведущий от исходного события к завершающему. Длина критического пути определяет время выполнения всего проекта.

Полный резерв работы - максимальное время, на которое можно увеличить продолжительность этой работы или участка некритического пути, не изменяя продолжительности критического пути.

Если полный резерв R_n используется полностью или частично, уменьшатся полные резервы других работ, принадлежащих этому участку некритического пути. Полный резерв работы определяется выражением:

$$R_n = t_j^{PO} - t_i^{PH} - t_{ij} \quad (5.2),$$

где t_j^{PO} - время позднего окончания предыдущей работы;

t_i^{PH} - время раннего начала текущей работы;

t_{ij} - продолжительность текущей работы.

Свободный резерв - максимальное количество времени, на которое можно увеличить продолжительность данной работы, не изменяя при этом ранних сроков начала последующих работ, при условии, что начальное событие данной работы произошло в свой ранний срок. Свободный резерв R_c определяется выражением

$$R_c = t_i^{PO} - t_i^{PH} - t_{ij} \quad (5.3)$$

где t_i^{PO} - время раннего окончания текущей работы;

t_i^{PH} - время раннего начала текущей работы.

Расчет ранних сроков начала работ ведут с начала графика по формуле:

$$t_j^{PH} = t_i^{PH} + t_{ij} \quad (5.4),$$

где t_i^{PH} - ранний срок начала предшествующей работы.

Расчет поздних сроков ведут с конца графика по формуле:

$$t_i^{PH} = t_j^{PH} - t_{ij} \quad (5.5),$$

где $t_j^{пн}$ - поздний срок начала последующей работы.

Если к одному событию подходит несколько работ, то при расчете раннего срока свершения события выбирают наибольшее значение. И, наоборот, при расчете поздних сроков свершения событий, если из одного события выходит несколько работ, то выбирают наименьшее значение.

Результаты расчетов представлены в таблице 5.5.

Таблица 5.5 - Расчеты времени

Шифр работ	t_{ij} , дней	$t_{ij}^{рн}$, дней	$t_{ij}^{ро}$, дней	$t_{ij}^{пн}$, дней	$t_{ij}^{по}$, дней	R_{ij}^n , дней
0-1	2	0	2	0	2	0
1-2	3	2	5	2	5	0
2-3	4	5	9	5	9	0
3-4	1	9	11	9	11	1
3-5	2	9	11	9	11	0
4-6	1	10	12	11	12	1
5-6	1	11	12	11	12	0
4-7	1	10	14	11	14	3
6-7	2	12	14	12	14	0
7-8	8	14	22	14	22	0
8-9	2	22	24	22	24	0
9-10	4	24	28	24	28	0
9-12	8	24	39	24	39	7
10-11	5	28	33	28	33	0

Шифр работ	t_{ij} , дней	t_{ij}^{PH} , дней	t_{ij}^{PO} , дней	$t_{ij}^{ПН}$, дней	$t_{ij}^{ПО}$, дней	R_{ij}^n , дней
11-12	6	33	39	33	39	0
12-13	4	39	43	39	43	0
13-14	3	43	46	43	46	0

Сетевой график представлен на рис. 5.1.

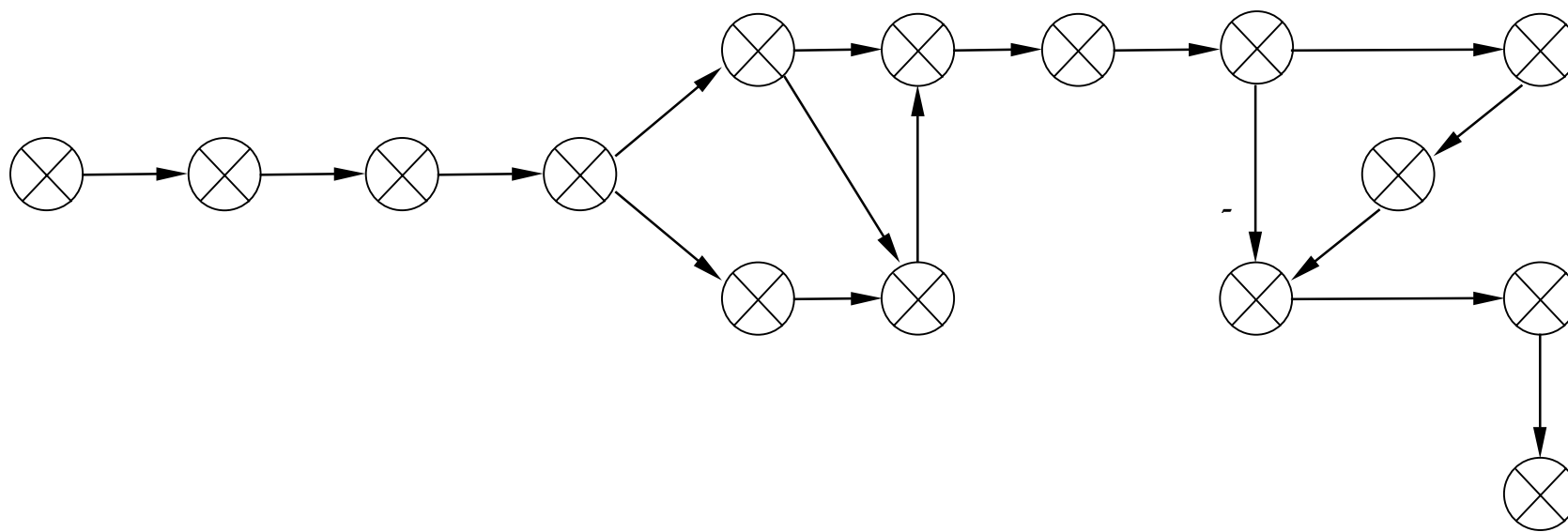


Рисунок 5.1 - Сетевой график

5.2.4 Карта загрузки исполнителей.

Карта загрузки исполнителей позволяет в наглядном виде представить потребность в специалистах той или иной квалификации при выполнении комплекса работ по проекту. Для построения карты загрузки исполнителей необходимо сначала произвести выбор и обоснование числа исполнителей и их квалификации.

Пусть в работе над проектом участвуют три специалиста различной квалификации:

- а) “руководитель” - специалист с самой высокой квалификацией из всех, участвующих в проекте;
- б) “инженер-программист - 1” - специалист высокой квалификации;
- с) “ инженер-схемотехник ” - специалист средней квалификации.

Занятость исполнителей при проведении комплекса работ по проекту представлена в таблице 5.6

Таблица 5.6 - Занятость исполнителей

Шифр работы	Привлекаемый исполнитель
0-1	Руководитель
1-2	Инженер-программист - 1
2-3	Инженер-программист - 1
3-4	Руководитель
3-5	Инженер-программист - 1

Шифр работы	Привлекаемый исполнитель
4-6	Инженер-схемотехник
5-6	Инженер-программист - 1
4-7	Руководитель
6-7	Инженер-схемотехник
7-8	Инженер-программист - 1
8-9	Инженер-программист - 1
9-10	Инженер-программист - 1
9-12	Инженер-схемотехник
10-11	Инженер-программист - 1
11-12	Инженер-программист - 1
12-13	Инженер-программист - 1
13-14	Руководитель

Карта загрузки исполнителей представлена на рис. 5.2.

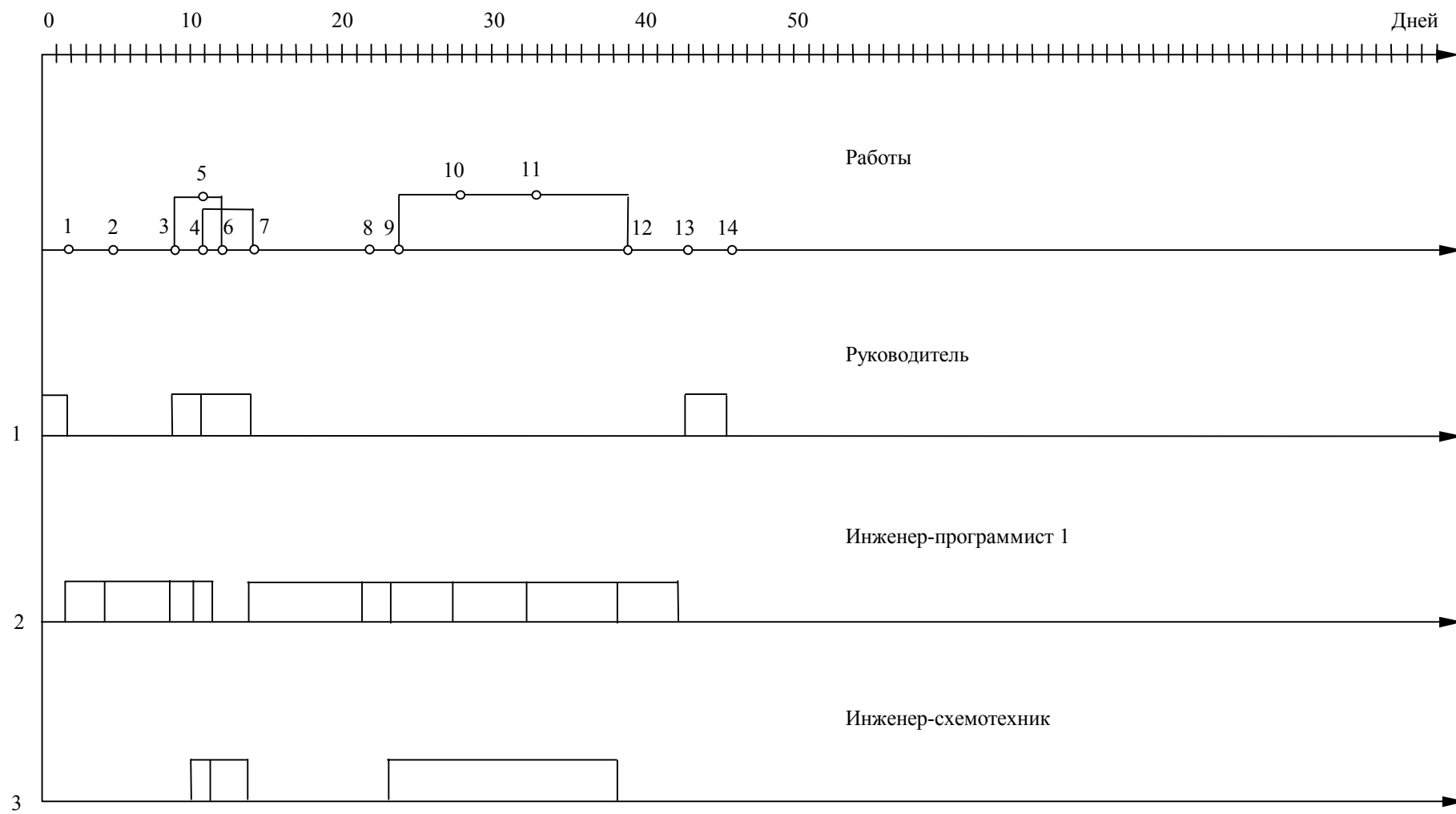


Рисунок 5.2 - Карта загрузки

5.2.5 Расчет затрат на разработку проекта

Работы производятся в сетевом отделе фирмы «Стек».

Затраты, связанные с проведением работ, включают в себя следующие расходы:

- 1) материалы основные;
- 2) специальное программное обеспечение для проведения исследований;
- 3) заработная плата исполнителей;
- 4) отчисления на социальные нужды;
- 5) налоги, относящиеся на себестоимость;
- 6) прочие прямые расходы;
- 7) накладные расходы.

Таблица 5.7- Затраты на основные материалы

Материалы	Кол-во	Ед. Измерения	Цена, тыс. руб	Сумма, тыс. руб.
1. Компьютер в комплекте	1	шт.	5000	5000
2. Основное программное обеспечение	1	шт.	100	100
Итого				5100

Таблица 5.8- Затраты на специальное программное обеспечение

Наименование	Кол-во	Цена, тыс. Руб	Сумма, тыс. руб.
1. Visual Basic Enterprise Edition Version 5.0	1	15330	15330
Итого			15330

Для расчета заработной платы исполнителей необходимы следующие данные:

- а) размер минимальной заработной платы - 83,49 тыс. руб.;
- б) размер ставки первого разряда единой тарифной сетки (ЕТС) - 83,49 тыс. руб.;
- с) должности, занимаемые исполнителями, и их научные степени для расчета должностных окладов и доплат за научную степень;
- д) районный коэффициент - 1,3;
- е) сроки занятости каждого исполнителя выполнением работ по проекту.

Будем считать, что

- а) руководитель - соответствует семнадцатой сетке ЕТС;
- б) инженер-программист - 1 - соответствует пятнадцатой сетке ЕТС;
- с) инженер-программист - 2 - соответствует двенадцатой сетке ЕТС.

Тогда месячная заработная плата руководителя с учетом доплат за должность и научную степень составит:

$$\begin{aligned} \text{ЗП}_{\text{рук}} &= (83,49_{\text{тыс. руб.}} \cdot 9,07 \cdot 1,6 + 5 \cdot 83,49_{\text{тыс. руб.}}) \cdot 1,3 = \\ &= 2117,8_{\text{тыс. руб.}} \end{aligned}$$

Заработная плата двух других исполнителей составит

$$\begin{aligned} \text{ЗП}_{\text{ин ж1}} &= (83,49_{\text{тыс. руб.}} \cdot 7,36 \cdot 1,4 + 3 \cdot 83,49_{\text{тыс. руб.}}) \cdot 1,3 = \\ &= 1444_{\text{тыс. руб.}} \end{aligned}$$

$$\text{ЗП}_{\text{ин ж2}} = 83,49_{\text{тыс. руб.}} \cdot 5,1 \cdot 1,3 = 553,5_{\text{тыс. руб.}}$$

Сроки занятости исполнителей отражены в таблице 5.9

Таблица 5.9- Занятость исполнителей

Исполнитель	Занят, дней
-------------	-------------

Исполнитель	Занят, дней
1. Руководитель, доктор, профессор	10
2. Инженер-программист - 1, совместитель, доцент, к. т. н.	39
3. Инженер-схемотехник	19

Размеры оплаты труда исполнителей представлены в
таблице 5.10.

Таблица 5.10 - Оплата труда исполнителей

Исполнитель	Месячная заработная плата, тыс. руб.	Срок занятости, дней	Размер оплаты труда, тыс. руб.
1. Руководитель	2117,8	10	882,4
2. Инженер-программист - 1	1444	39	2346,5
3. Инженер-схемотехник	553,5	19	438,2
Итого (ФОТ)			3667,1

Размеры отчислений на социальные нужды и налоги, относящиеся на себестоимость составляют (в % от ФОТ):

- пенсионный фонд - 28%;
- фонд социального страхования - 5,4%;
- фонд обязательного медицинского страхования - 3,6%;
- фонд занятости - 1,5%;
- транспортный налог - 1%;
- налог на нужды образовательных учреждений - 1%.

Составим смету затрат.

Таблица 5.11- Смета затрат

Наименование статьи затрат	Сумма, тыс. руб.
1. Основные материалы	5100
2. Специальное программное обеспечение	15330
3. Оплата труда (ФОТ)	3667,1
4. Отчисления на социальные нужды и во внебюджетные фонды	
– пенсионный	1026,8
– фонд социального страхования	198
– фонд обязательного медицинского страхования	132
– фонд занятости	
– транспортный налог	55
– налог на нужды социальных учреждений	36,7
Итого	36,7
	1485,2
5. Прочие прямые расходы (5% от предыдущих)	1279,115
6. Накладные расходы (20% от прямых затрат)	5372,283
Итого	35058,498

Определим договорную цену, считая размер прибыли равным 20% и НДС равным также 20%.

$$\begin{aligned}
 \text{ДЦ} &= (\text{ОЗ} + \text{Пр}) + \text{НДС} = \text{ОЗ} * 1,2 * 1,2 = \\
 &= 35058,498 \text{ т.р.} * 1,2 * 1,2 = 50484,23712 \text{ т.р.}
 \end{aligned}$$

6. Вопросы охраны труда

6.1. Анализ объективных факторов производственной опасности

Во время проведения разработок по теме диплома не раз затрагивались вопросы безопасности жизнедеятельности. Выделим основные факторы риска при проведении работ. Комплекс СМ-600 который подвергался доработке расположен на территории цеха по изготовлению печатных плат в котором сочетаются как механические, так и химико-гальванические работы. Для предотвращения ситуаций опасных для жизни и здоровья человека в ПО 'Контур' существует Стандарт Предприятия в соответствии с которым разработаны Инструкции по охране труда. Ниже приводится несколько таких документов для различных специальностей.

Инструкция по технике безопасности для сверловщиков.

- 1.1 К работам на сверлильных станках могут быть допущены только лица, обученные сверлильному делу и прошедшие инструктаж по ТБ на рабочем месте. При проведении инструктажа на рабочем месте сверловщику следует показать безопасные приёмы и методы работы.
- 1.2 На каждую работу сверловщику должна выдаваться технологическая карта, предусматривающая применение определённых видов инструмента, приспособлений и т.д. В технологической карте следует предусматривать необходимые мероприятия по ТБ. При выполнении разовых работ необходимо руководствоваться технологическими указаниями администрации цеха.
- 1.3 Инструмент, средства крепления, подъёмные механизмы и т.д. должны быть в исправном состоянии. Для удаления со станка металлической стружки должны выдаваться специальные щётки, крючки или применяться отсосы.
- 1.4 Применение самодельных и неисправных приспособлений, инструмента и т.д. строго запрещается.
- 1.5 У станков на рабочих местах должны находиться исправные рациональные сиденья и деревянные подножные решётки.

- 1.6 Рабочее место и проходы у станков необходимо содержать в чистоте и порядке. Заготовки и детали должны храниться в специальной таре или укладываться в отведённых местах на подкладках в штабеле не более 1м. Нельзя допускать на рабочих местах большого скопления металлической стружки и отходов производства.
- 1.7 Станки должны быть оборудованы местным низковольтным освещением напряжением 36 В.
- 1.8 В станках необходимо предусмотреть ограждение шпинделя и предохранительные устройства, выключающие подачу шпинделя при его перегрузке.
- 1.9 Устанавливать на станок и снимать со станка детали, приспособления и инструмент, вес которых раздельно превышает 16 кг, необходимо при помощи подъёмных механизмов и приспособлений /кранов и т.д./
- 1.10 Станки, на которых в процессе обработки материалов /бронза, чугун и т.д./ образуется пыль должны быть оборудованы устройствами для её удаления.
- 1.11 Обрабатываемые изделия следует устанавливать и закреплять в тисках, зажимах и др. Приспособлениях, надёжно укреплённых на столе или плите сверлильного станка.
- 1.12 Смена свёрл и др. Инструмента разрешается только при остановке станка и с применением безопасных, быстросменных патронов.
- 1.13 Сверлильные станки необходимо снабдить клиньями для крепления оправок в шпинделе.
- 1.14 Для сверления вязких металлов целесообразно применять спиральные свёрла со стружкообразным профилем.
- 1.15 Одновременная работа двух рабочих на двухшпиндельном станке запрещается.

2. Обязанности сверловщика.

До начала работы.

- 2.1 Приведи в порядок свою одежду: застегни обшлага, убери волосы под головной убор или под косынку без свисающих концов. Выпускать волосы из-под косынки запрещается.

2.2 Детали уложи в тару или на подкладки устойчиво и чтобы они не загромождали рабочее место и проходы.

2.3 Осмотри станок и убедись в исправности механизмов управления, заземления, ограждений и пусковых устройств и т.д.

2.4 Приведи в порядок рабочее место, ненужные для работы предметы убери, проверь исправность сидения.

2.5 В случае неисправности станка сообщи мастеру и без его разрешения не приступай к работе.

Во время работы.

2.6 Для установки на станок тяжёлых кондукторов, приспособлений, деталей пользуйтесь подъёмными устройствами.

2.7 Применяй исправные ключи по размерам головок болтов и гаек. Наращивание ключей трубами запрещается.

2.8 Надёжно закрепляй обработанную деталь, материал и инструмент. Удерживать детали, материалы руками при сверлении запрещается.

2.9 Правильно устанавливай и надёжно закрепляй сверло в патроне станка.

2.10 Не тормози станок нажимом руки или случайным предметом на вращающийся шпиндель или сверло.

2.11 Работай на станке в защитных очках.

2.12 Охлаждай сверло с помощью кисточки или капельницы. Выполнять это тряпками или концами запрещается.

2.13 при сверлении спиральными свёрлами образующуюся стружку убирай щёткой или крючком. Сдувать и выдувать стружку из отверстий сжатым воздухом или ртом запрещается.

2.14 Не работай на станке в рукавицах или перчатках и с забинтованными пальцами без резиновых напальчников. При образовании стружки не допускай наматывания её на сверло или деталь. Удаляй стружку только с помощью крючка типа 'Рапира'.

2.15 В случае заедания сверла в детали немедленно останови станок.

2.16 Укладывать детали и инструменты на столе станка запрещается.

2.17 Останавливай станок в следующих случаях:

- а/ даже при кратковременном уходе с рабочего места
- б/ при чистке, смазке, уборке станка
- в/ при установке или перестановке крепёжных устройств и деталей
- г/ при измерении отверстий и смене сверла
- д/ при обнаруженных во время работы неисправностях

2.18 При временном прекращении э/энергии, переключи в нейтральное положение передачу станка и выведи сверло из детали.

2.19 Хранить рабочий инструмент, детали и личные вещи в ящиках э/щитов и др. Местах вблизи токоведущей части строго запрещается.

2.20 О произошедшем несчастном случае с тобой или товарищем по работе немедленно поставь в известность мастера, начальника цеха.

По окончании работы.

2.21 Останови станок, и выключи э/двигатель, после чего убери рабочее место.

2.22 Очистить станок, приспособления и ограждения от грязи и стружки. смажь станок и сдай его и рабочее место своему сменщику и мастеру в чистоте и порядке.

2.23 Убери приспособления в отведённые для них места, инструмент в инструментальный ящик /шкафчик/, причём укладывая свёрла хвостовиком к себе.

2.24 О всех неисправностях станка и приспособлений сообщи мастеру и сменщику.

4. Правила противопожарной безопасности.

4.1 Ответственность за пожарную безопасность на производственном участке несёт мастер.

4.2 Должностные лица, ответственные за пожарную безопасность обязаны:

4.2.1 Обеспечить соблюдение противопожарного режима.

4.2.2 Следить за исправностью отопления, вентиляции, электроустановок, принять меры к устранению обнаруженных неисправностей, могущих привести к пожару.

- 4.2.3 Обеспечить исправное содержание и готовность имеющихся средств пожаротушения, связи и сигнализации.
- 4.2.4 следить за тем, чтобы после окончания работы проводилась уборка помещения.
- 4.2.5 Промасленные обтирочные материалы и производственные отходы должны храниться в специально отведённых местах.
- 4.2.6 Запрещается использовать пожарную технику для хозяйственных нужд.
- 4.2.7 Каждый рабочий, обнаруживший пожар или загорание обязан сообщить мастеру, начальнику цеха и приступить к ликвидации пожара.

5. Ответственность

5.1 Лица, нарушившие настоящую инструкцию в зависимости от степени и характера нарушений привлекаются к дисциплинарной, материальной и общественной ответственности.

5.1.1 Дисциплинарная ответственность применяется в связи с тяжестью поступка и когда нельзя применить воздействие коллектива. Налагаются следующие виды взысканий:

замечание

выговор

строгий выговор

перевод на нижеоплачиваемую работу

увольнение

5.1.2 Основанием для привлечения к материальной ответственности является прямой фактический ущерб, независимо от привлечения к другим видам ответственности в пределах 1/3 месячной тарифной ставки работника.

Далее приведём некоторые выдержки из 'Инструкции по технике безопасности для рабочих гальванического участка цеха №11'.

1. К работе на гальваническом участке допускаются лица, достигшие 18-летнего возраста, прошедшие инструктаж по технике безопасности и противопожарной безопасности.

2. На участке всегда должна быть аптечка, содержащая всегда полный набор медикаментов. На внутренней стороне дверцы аптечки должна иметься опись медикаментов и средств для оказания первой помощи.
3. Помещение должно быть обеспечено противопожарным инвентарём, огнетушителем, ящиком с песком.
4. Все токоведущие части оборудования должны быть заземлены.
5. Нельзя находиться на участке с выключенной вентиляцией.
6. Не хранить на участке пищевые продукты.
7. За 30 минут до начала работы проветрить помещение, включить вентиляцию.
8. Не принимать пищу и не курить в помещении, где производится гальваническое покрытие.
9. Перед едой и курением (в местах, специально отведённых для этого) обязательно вымыть руки с мылом и прополоскать рот.

Одним из видов работ при написании данной работы является программирование на персональной ЭВМ. Так как во время разработки программ для персонального компьютера человек находится в непосредственном соприкосновении со средствами вычислительной техники, необходимо проработать вопросы безопасности при работе с ЭВМ. В соответствии с этим можно выделить следующие факторы производственной опасности :

- Питание ЭВМ осуществляется от сети переменного напряжения 220 В, 50 Гц;
- ЭВМ содержит в себе вентиляторы, накопители на жестких и гибких дисках, которые являются источниками шума;
- ЭВМ может содержать источники высокочастотных электромагнитных излучений;
- Монитор ЭВМ является источником слабого рентгеновского и низкочастотного электромагнитного излучения;

6.2. Требования безопасности при работе с ЭВМ

6.2.1. Уровни шума и звуковой мощности в местах расположения персонала АСУ не должны превышать значений, установленных ГОСТ 27818-88 и санитарными нормами, при этом должны быть учтены уровни шумов и звуковой мощности, создаваемые всеми источниками, в том числе и акустическими средствами передачи данных. ГОСТ 27818-88 (СТ СЭВ 5147-85) «Допустимые уровни шума» устанавливает допустимый уровень шума на рабочих местах при эксплуатации технических средств, вычислительных машин и систем обработки данных, характеризующихся эквивалентным уровнем звука A .

Устройства следует устанавливать таким образом, чтобы при эксплуатации их на рабочих местах значения эквивалентного уровня звука A (L_{Aeq}) не превышали допустимых значений, приведенных в таблице 6.1.

Таблица 6.1 - Допустимые уровни звука

Категория рабочего места		До пустимые значения L_{Aeq} , дБ	Примеры устройств, которые снабжены указанные рабочие места
Рабочие места устройств подготовки данных в вычислительных центрах	на бумажных носителях данных	до 75	Устройство перфорирования карт, устройство перфорирования лент
	на магнитных носителях данных	до 60	Накопители на магнитных дисках, накопители на магнитной ленте
Рабочие места обслуживания устройств в вычислительных центрах		до 70	Пульт оператора, центральный процессор

Рабочие места с использованием устройств в административных помещениях и лабораториях, связанных с повторяющимися операциями	до 60	ЭВМ для проведения конторских и коммерческих работ
Рабочие места с использованием устройств для исследований, разработки, конструирования, программирования и врачебной деятельности	до 50	Дисплей, клавиатуры, настольные вычислительные машины

6.2.2 Общие эргономические требования.

Общие эргономические требования к рабочим местам персонала АСУ устанавливаются в соответствии с ГОСТ 22269 - 76 «Рабочее место оператора. Взаимное расположение элементов рабочего места». Настоящий стандарт распространяется на индивидуальные рабочие места операторов стационарных и подвижных объектов системы «человек машина» и устанавливает общие эргономические требования к взаимному расположению элементов рабочего места: пульта управления, средств отображения информации, органов управления, кресла, вспомогательного оборудования.

6.2.2.1. Общие требования ГОСТа :

а) При взаимном расположении элементов рабочего места необходимо учитывать:

- рабочую позу человека-оператора;
- пространство для размещения человека-оператора;

- возможность обзора элементов рабочего места;
- возможность обзора пространства за пределами рабочего места;
- возможность ведения записей, размещения документации и материалов, используемых человеком-оператором.

б) Взаимное расположение элементов рабочего места должно обеспечивать возможность осуществления всех необходимых движений и перемещений для эксплуатации и технического обслуживания оборудования. При этом должны учитываться ограничения, налагаемые спецодеждой и снаряжением человека-оператора.

в) Взаимное расположение элементов рабочего места должно обеспечивать необходимые зрительные и звуковые связи между оператором и оборудованием, а также между операторами.

г) При расположении элементов рабочего места должны быть предусмотрены необходимые средства защиты человека-оператора от воздействия опасных и вредных факторов, предусмотренных ГОСТ 12.0.003 - 74 «Опасные и вредные производственные факторы», а также условия для экстренного ухода человека-оператора с рабочего места.

д) Взаимное расположение элементов рабочего места должно способствовать оптимальному режиму труда и отдыха, снижению утомления оператора, предупреждению появления ошибочных действий.

е) Взаимное расположение пульта управления, кресла, органов управления и средств отображения информации должно производиться в соответствии с антропометрическими показателями, структурой деятельности, психофизиологическими и биомеханическими характеристиками человека-оператора.

6.2.2.2 Требования к размещению органов управления :

а) При размещении органов управления необходимо выполнять следующие эргономические требования:

- органы управления должны располагаться в зоне досягаемости моторного поля;

- наиболее важные и часто используемые органы управления должны быть расположены в зоне легкой досягаемости моторного поля;
- органы управления, связанные с определенной последовательностью действий оператора, должны группироваться таким образом, чтобы действия оператора осуществлялись слева направо и сверху вниз;
- расположение функционально идентичных органов управления должно быть единообразным на всех панелях рабочего места;
- расположение органов управления должно обеспечивать равномерность нагрузки обеих рук и ног человека-оператора;

б) Органы управления и функционально связанные с ними индикаторы необходимо располагать вблизи друг друга функциональными группами таким образом, чтобы орган управления или рука оператора при манипуляциях с ним не закрывали индикатора. При этом органы управления должны располагаться в соответствии с последовательностью действий, выполняемых оператором.

в) Органы управления, применяемые только для технического обслуживания и регулировки, должны размещаться отдельно от остальных органов управления или быть изолированными от человека-оператора на период выполнения им основной работы.

6.2.2.3 Требования к размещению средств отображения информации :

а) Средства отображения информации необходимо группировать и располагать группы относительно друг друга в соответствии с последовательностью их использования или с функциональными связями элементов систем, которые они представляют. При этом средства отображения информации необходимо размещать в пределах групп так, чтобы последовательность их использования осуществлялась слева направо или сверху вниз.

б) Лицевые поверхности индикаторов следует располагать в оптимальной зоне информационного поля в плоскости, перпендикулярной нормальной линии взора оператора, находящегося в рабочей позе. Допускаемое отклонение

от этой плоскости - не более 45'; допускаемый угол отклонения линии взора от нормальной - не более 25' для стрелочных индикаторов и 30' для индикаторов с плоским изображением.

6.3 Разработка мер безопасности на рабочем месте оператора ЭВМ

6.3.1 Все внешние элементы технических средств АСУ, находящиеся под напряжением, должны иметь защиту от случайного прикосновения, а сами технические средства, иметь зануление или защитное заземление в соответствии с ГОСТ 12.1.030 - 81 и «Правилами устройства электроустановок».

6.3.2 Уровни шума и звуковой мощности в местах расположения персонала АСУ не должны превышать значений, установленных ГОСТ 27818-88 (СТ СЭВ 5147-85) «Допустимые уровни шума» и санитарными нормами, при этом должны быть учтены уровни шумов и звуковой мощности, создаваемые всеми источниками, в том числе и акустическими средствами передачи данных.

6.3.3 Уровни освещенности рабочих мест персонала АСУ должны соответствовать характеру и условиям труда. Должны быть предусмотрены защита от слепящего света и устранение бликов.

6.4 Разработка инструкции по безопасности на рабочем месте оператора ЭВМ (Выдержки из инструкции М 260-93 «По охране труда для операторов, пользователей, и при работе с внешними устройствами электронно-вычислительной техники от 16.07.93 г»)

6.4.1 Общие положения

6.4.1.1 Инструкция распространяется на операторов, пользователей и персонал, работающий с внешними устройствами ЭВМ.

6.4.1.2 Работники, в соответствии с приказом Минздрава СССР М 555 от 1989 г., в целях предупреждения у них профессиональных заболеваний должны проходить периодические медицинские осмотры (1 раз в два года).

6.4.1.3 При работе с ЭВМ возможны следующие основные опасности:

- поражение электрическим током;

- получение травм от движущихся частей внешних устройств;

В зависимости от конструктивных особенностей, устройства визуального отображения генерируют несколько типов излучения, в том числе: рентгеновское, радиочастотное и ультрафиолетовое. Если не выполнять профилактические мероприятия и не соблюдать режим работы, работа с ЭВМ, как правило, сопровождается значительными зрительным и общим переутомлением.

6.4.1.4 Организация рабочих мест.

Рабочий стол должен регулироваться по высоте в пределах 680-760 мм, при отсутствии такой возможности его высота должна составлять 720 мм. Оптимальные размеры столешницы должны составлять 1600х900 мм, Под столешницей рабочего стола должно быть свободное пространство для ног с размерами по высоте не менее 600 мм, по ширине -500 мм, по глубине-650 мм.

Размер экрана должен быть не менее 31 см (14 дюймов) по диагонали, при этом расстояние от глаз до экрана должно быть в пределах 40-80 см.

6.4.1.5 Режим труда и отдыха

В соответствии с особенностью трудовой деятельности в режиме труда должны быть дополнительно введены 2-3 регламентированных перерыва длительностью 10 мин каждый; два перерыва - при 8-ми часовом рабочем дне и три перерыва - при 12-ти часовом рабочем дне.

Непрерывная продолжительность работы с видеотерминалом не должна превышать четырех часов при 8-ми часовом рабочем дне, через каждый час работы необходимо вводить перерыв на 10 - 15 мин, а через два часа - 15 мин.

В целях профилактики переутомления и перенапряжения при работе с дисплеями необходимо выполнять во время регламентируемых перерывов комплексы специальных упражнений (они приведены ниже).

С целью снижения или устранения нервно-психического зрительного и мышечного напряжения, предупреждения переутомления рекомендовать в специально оборудованном помещении проводить сеансы психофизиологической разгрузки и снятия усталости во время регламентированных перерывов и после окончания рабочего дня.

6.4.2 Обязанности персонала при аварийных ситуациях.

При работе на устройствах возможны следующие аварийные ситуации:

- попадание корпуса устройства под напряжение при коротком замыкании в электросистеме, пробое изоляции и отсутствии или неисправности заземления;
- поломки отдельных узлов устройства от перегрузки, нарушения правил пользования;

Во всех аварийных ситуациях необходимо прекратить работу, обесточить устройства. До устранения неисправности пользоваться устройствами запрещено.

Заключение

В ходе данной работы было изучено состояние развития программных продуктов для автоматизации проектирования печатных плат. Создана работоспособная технология проектирования ПП и вывод технологической документации на изделия. Разработаны методы адаптации зарубежных программных комплексов к отечественной технологии конструирования ПП

В результате проведённых работ создано устройство сопряжения отечественного автоматизированного сверлильного комплекса СМ-600 с персональным компьютером посредством последовательного асинхронного порта RS232. Разработаны программы по передачи сверловочных данных в стандартном формате EXELLON (от различных программ САМ) в комплекс СМ-600.

В процессе работы написаны вспомогательные программы для сопряжения персонального компьютера типа IBM PC с фотоплоттером типа ЭММА 30 из комплекса QWEST, создано устройство для подключения перфоввода типа 'Консул' к компьютеру и разработано программное обеспечение для ввода и архивирования информации с перфолент. Решена проблема преобразования информации от программного комплекса КЛАРА в стандартный формат EXXELON.

В дальнейшем предполагается полный перевод подразделений завода на разработанную технологию проектирования печатных плат.

Список использованных источников

1. Фаронов В.В. Турбо Паскаль. Книга 1. Основы Турбо Паскаля. М. 1992
2. Климов Ю.С. Касаткин А.И. Мороз С.М. Программирование в среде TURBO PASCAL 6.0: Справ. пособие. -Мн.: Выш.шк., 1992
3. Касаткин А.И. Профессиональное программирование на языке Си. Системное программирование. - Мн.: Выш.шк., 1993. - 301 с.: ил.
4. Мячев А.А., Степанов В.Н. Персональные ЭВМ и микроЭВМ. Основы организации: Справочник/ Под ред. А.А. Мячева. - М.: Радио и связь, 1991. - 320 с.: ил.
5. IEEE 1284 - Updating the PC Parallel Port.
6. Сучков Д.И. Адаптация САПР PCAD к отечественному технологическому оборудованию: Программирование в САПР PCAD. -Обнинск: "Призма", 1993. - 460с.
7. Скэнлон Л. Персональные ЭВМ IBM PC и XT. Программирование на языке ассемблера: Пер. с англ. - 2-е изд., стереотип. - М.:Радио и связь. 1991.-336с.: ил.
8. Сверлильно-фрезеровальный станок СМ-600. Техническая документация.

Приложение А

Текст программы MAINDRL.PAS

```
uses dos,crt;
```

```
const
```

```
clr      : byte = $0e;           { цвет надписей }
clrbr    : byte= $0f;           { яркий цвет }
delay_const : word = 1;         { задержка между передачей байт (в ms) }
startline : string = '%';       { символ начала перфоленты }
firstline  : string = 'G27G81G90R3W-5F1600S40'; { первая линия с начальными
установками }
end1line   : string = 'G80T00M05'+#$0a+'M02';
end2line   : string = 'E0X0Y0Z0';
toolline   : string = 'F1600S40'; { смена инструмента }
perfoendline: string = #$99;     { символ конца перфоленты }
comnow     : byte = 1;          { номер COM порта (0 это первый) }
```

```
helptxt:array[1..5] of string[41] = (
'Программа эмуляции устройства считывания',
'с перфоленты УСЛ-300 ',
'формат вызова:   main_drl.exe file',
'file - имя файла в формате станка EXELLON',
'Автор Сараев К.В.      2:5005/14.34'
);
```

```
var
```

```
f      : text;                  { просматриваемый файл }
ba     : word;                  { базовый адрес порта }
a      : byte;                  { считанный байт из файла }
prttmp : byte;                  { копия содержимого порта управления в памяти }
posit  : longint;               { число переданных байт }
allbyte : longint;              { кол-во байт которое нужно передать }
allstring:longint;
metric: boolean;               { в какой системе исходные данные }
session_failure : boolean;
diam   : array [1..9] of real;  { массив с диаметрами инструментов }
hole   : array [1..9] of longint; { массив с числом отверстий каждого диаметра }
firststring : boolean;         { признак того что начальные установки ещё не посылались }

r      : registers;
s      : string;
i:byte;
```


pst:string;

u:boolean;

label hhh;

```
procedure error(s:string); { выводит сообщение об ошибке s и выходит из программы }
begin
write(s);
exit;
end;
```

```
function tobin(a:byte):string; { преобразует байт a в его битовое изображение }
var i:byte;
s:string;
begin
s:='';
for i:=7 downto 0 do { формируем строчку }
if (a and trunc(exp(i*ln(2))))<>0 then s:=s+'*' else s:=s+'-';
tobin:=s;
end;
```

```
function scale(a,b:longint;n:byte):string; { формируют шкалу длиной n символов }
var { заполненную на a/b части }
e:real;i:byte;
s:string;
begin
s:='';
e:=round(a/b*n);
for i:=1 to n do
if i>e then s:=s+'-' else s:=s+'.';
scale:=s;
end;
```

```
procedure color(c:byte; inv: boolean); { устанавливает цвет/инверсный цвет }
var f,m : byte;
begin
m:=c; f:=c;
if inv then c:=(f shr 4)or(lo(m shl 4));
textcolor(c and $0f);
textbackground(c shr 4);
end;
```

```
procedure cursor(f:boolean); { включает или выкл. курсор }
begin
```

```

r.ah:=1;
if f=true then r.cx:=$0e0f else r.ch:=$20;
intr($10,r);
end;

```

```

function tonum(s:string;a:char):real; { выдает численное значение метки A }
var i,p,rr:integer;
r:real;
begin
i:=1;
while i<=ord(s[0]) do { вставляем пробелы между командами }
  if s[i]>'9' then begin insert(' ',s,i); inc(i,2); end else inc(i);
  s:=s+' '; { вставляем заключительный пробел }
p:=pos(a,s);
delete(s,1,p);
s:=copy(s,1,pos(' ',s)-1);
val(s,r,rr);
if rr=0 then tonum:=r else tonum:=0;
end;

```

```

function oddd(a:byte): boolean;
var i,n,s:byte;
begin
n:=0;
for i:=0 to 7 do
  if (a and trunc(exp(i*ln(2))))<>0 then inc(n); { возводим 2 в степень n }
  if (n and 1)<>0 then oddd:=true else oddd:=false;
end;

```

```

procedure init; { определяет диаметры инструментов, число байт к перед }
}
var a:byte; s:string;
toolnow:byte; { текущий рассматриваемый инструмент }
begin
firststring:=true; { признак необходимости послыки строки с установками }
posit:=0;
for a:=1 to 9 do begin diam[a]:=0; hole[a]:=0; end;
allstring:=5;
allbyte:=length(startline)+length(firstline)+length(end1line)+length(end2line)+length(perfoen
e)+ 5 ;
reset(f); { выходим на начало файла }
repeat readln(f,s) until (pos('/DBUNIT',s)<>0) or (eof(f)); { ищем начало данных }
readln(f,s); { пропускаем M7x }
while not eof(f) do begin { начинаем просматривать файл }
  readln(f,s);
  if s='M30' then exit; { просмотрели весь файл }

```

```

if pos('T',s)<>0 then begin      { встретилась смена инструмента }
    diam[trunc(tonum(s,'T'))]:=tonum(s,'C');    { считываем его диаметр }
    toolnow:=trunc(tonum(s,'T'));      { изменяем текущий инструмент }
    if firststring=true then begin { если смена инструмента включается в первую строку }
        inc(allbyte,2);
        firststring:=false; end
    else begin
        inc(allbyte,3+length(toolline));    { учитываем смену в числе передаваемых байт }
        inc(allstring);
        end
    end
    else begin
        inc(allbyte,length(s)+1);          { иначе учит. длину определения координат }
        inc(allstring);
        inc(hole[toolnow]);                { увеличиваем число отверстий }
        end;
    end;
end;

```

```

procedure say(b:byte);
var r:registers;
begin
    r.dx:=comnow;
    r.ah:=1;
    r.al:=b;
    intr($14,r);
end;

```

```

function get_tx(var b:byte):boolean;
var r:registers;
begin
    r.dx:=comnow;
    r.ah:=2;
    intr($14,r);
    b:=r.al;
    get_tx:=true;
end;

```

```

procedure send_str(s:string);
var i:byte;
begin
    for i:=1 to length(s) do { пока не кончилась строка кидаем байты в порт }
        say(ord(s[i]));
    end;

```

```

procedure sendstring(s:string);

```

```

var i:byte;
begin
if session_failure then exit;
s:=s+#$0a;      { вставляем конец кадра }
send_str('ss'+#$0d); delay(100);
say(length(s));
for i:=1 to length(s) do begin  { пока не кончилась строка кидаем байты в порт }
a:=ord(s[i]);
if odd(a and $7f) then a:=a or $80;  { выставляем четность }
say(a);
end;
inc(posit);
gotoxy(30,3); write(scale(posit,allstring,40));      { шкала процесса }
gotoxy(10,3); write(posit:6);
repeat      { ждём пока контроллер передаст строку }
gotoxy(1,6);textcolor(clrbr); write('Ждём Подтверждение'); color(clr,false);
if keypressed then
  if ord(readkey)=$1b {esc} then begin      { останов по ESC }
    gotoxy(1,6); write('');
    gotoxy(30,12); write('Выход из программы (Y/N) ? ');
    cursor(true); pst:=readkey;
    if (pst='Y') or (pst='y') then halt;
    cursor(false);
    gotoxy(30,12); write('');
  end;
until (get_tx(a)) and ((a=82)or(a=70));
if a=70 then session_failure:=true;
gotoxy(1,6); write('');
gotoxy(5,20);write(a);
end;

```

```

function ready_in:boolean;
var a:byte;
begin
send_str('status'+#$0d);
repeat until get_tx(a);
gotoxy(1,11);if (a and 1)<>0 then highvideo else lowvideo; write('ввод');
gotoxy(1,12);if (a and 2)<>0 then highvideo else lowvideo; write('пуск');
if a=$0ff then ready_in:=true else ready_in:=false;
end;

```

```

begin
pst:=ParamStr(1);
pst:='test.drl';
if (pst='/?') or (pst='') then begin      { заказали просмотр подсказки или не указали файл
writeln("");

```

```

for i:=1 to 19 do begin
  write(' ');
  writeln helptxt[i]);end;
end
else begin
  assign(f,pst); { инициализируем файл }
  {$i-}reset(f); {$i+}
  if ioresult<>0 then begin
    write('Файл ',pst,' не найден ');exit; end; { такой файл не найден }
  clrscr;
  cursor(false);
  ba:= memw[$40:$02]; { узнаем адрес порта COM2 }
  if ba=0 then error('Отсутствует адаптер последовательной связи COM2');

  r.dx:=1;
  r.ah:=0; { инициализируем }
  r.al:=$a3; { 2400 8N1 }
  intr($14,r);

  color(clr,false);
  gotoxy(1,1); write('Файл: ',pst:12);
  repeat
    session_failure:=false;
    init;
    firststring:=true; { признак необходимости послылки строки с установками }
    reset(f); { выходим на начало файла }
    gotoxy(14,8); for i:=1 to 9 do if diam[i]=0 then write(' ') else
      write(diam[i]:4:2,' '); { распечатываем диаметры }
    gotoxy(14,9); for i:=1 to 9 do if hole[i]=0 then write(' ') else
      write(hole[i]:4,' '); { распечатываем количество отверстий }
    gotoxy(1,7); write('Инструмент 1 2 3 4 5 6 7 8 9');
    gotoxy(1,8); write('Диаметр');
    gotoxy(1,9); write('Отверстий');
    a:=0;

    repeat readln(f,s)
    until (pos('/DBUNIT',s)<>0) or (eof(f)); { ищем начало данных }
    readln(f,s);
    if s='M71' then metric:=true else metric:=false; { определяем систему единиц }

    gotoxy(1,4); write('Всего ',allstring:6); { сколько надо передать байт }

  { repeat
    until ready_in; { ждём готовность устройства }

```

```

gotoxy(1,3); color(clrbr,true); write('Передано '); color(clr,false);

sendstring(startline);           { посылаем признак начала программы }

while not eof(f) do begin        { начинаем просматривать файл }
  readln(f,s);
  if s='M30' then goto hhh;      { просмотрели весь файл }
  if pos('T',s)<>0 then begin     { встретилась смена инструмента }
    if firststring=true then begin { впервые вставляем инструмент => посылаем начальную
установки }
      sendstring(firstline+copy(s,pos(s,'T'),2));
      firststring:=false; end
    else sendstring(copy(s,pos(s,'T'),2)+toolline);
    end
  else sendstring(s);            { встретилось определение координат }
  end;
hhh:
  sendstring(end1line);          { посылаем заключительные строки }
  sendstring(end2line);
  sendstring(perfoendline);
  gotoxy(30,12); write('Данные переданы. Для повтора нажмите клавишу. ');
  readkey;
  gotoxy(30,12); write('
');

until false;
end;
end.

```

Приложение Б

Текст программы DRILL51.ASM

```
PCON:          equ    87h
max_string     equ    15d ; максимальная длина принимаемой строки (остальное
обрезается)

mystack        equ    65h ; начало стека
rx_len         equ    60h ; длина принятой строки
rx_buf_len     equ    61h ; длина данных в буфере
rx_str         equ    30h ; место расположения принимаемой строки
rx_buf         equ    40h ; принятые данные для обработки соотв. прог. (rx_str
без команды)

str_ok         equ    20h.0 ; рассматриваемая строчка найдена в таблице
sym_was        equ    20h.1 ; ведущие пробелы кочились и был хоть один символ

main_lamp      equ    p2.0 ; сведодиод светиться, когда контр-р ожидает команду
                ; и гаснет когда переходит на её выполнение

NCstrobe       equ    p2.7 ; выходы для подключения ЧПУ
NCstart        equ    p2.6
NCinput        equ    p2.5
MCready        equ    p2.4 ; готовность контроллера (к ЧПУ)

    org    0h
    ljmp   start        ; переход на начало программы

    org    23h
    jb     ri, rxi       ; не пускаем прерывания от передатчика
    reti

rxi:  jmp   rx_int       ; п/п прерывания по приёму байта RS232

    org    30h
ctable: db    'help',0
        ajmp  help
        db    'load',0
        ajmp  load
        db    'send',0
        ajmp  send
        db    'ss',0
        ajmp  smallsend
        db    'status',0
```

```

        ajmp status

        nop                ; end for ctable

helptable: db 'for help',0
            ajmp help
            db 'for load',0
            ajmp load

        nop                ; end for ctable

help_str:  db '      *** DRILL ATTACHER ***',0dh,0dh
            db '      help file',0dh
            db 'Контроллер для подключения ЧПУ станка CM600',0dh
            db 'к COM порту компьютера IBM',0dh
            db '      Команды:',0dh
            db ' help - этот текст',0dh
            db ' load <данные> - передача <данных> в станок ',0dh
            db '      (кроме кода 0)',0dh
            db ' ss - передача данных в станок',0dh
            db '      после этой команды передаётся байт кол-ва',0dh
            db '      передаваемых данных и затем сами данные',0dh
            db 'Каждая команда заканчивается кодом 0dh',0dh
            db 'ВНИМАНИЕ! Компьютер должен быть заземлён',0dh
            db ' иначе передача данных будет невозможна',0dh,0

command_err: db ' - bad command (type "help")',0dh,0
load_str:    db 'LOAD ',0

rx_int:
rx1: push acc
    push psw
    mov psw,#00001000b        ; включаем второй банк регистров
    clr ri
    mov a,sbuf                ; считываем принятый байт

    ; mov r1,a
    ; call tx_strbyte          ; передаём принятый байт в символьном виде
    ; mov a,r1

```



```

        jb     sym_was,rx3          ; если уже принимали символы <> пробелу то
пробел не отслеживаем
        cjne  a,#' ',rx3          ; отслеживаем ведущие пробелы
        jmp   to_main              ; и не записываем их в память
rx3:  setb   sym_was                ; если сюда дошли значит ведущие пробелы
кочились
        cjne  r0,#(max_string+rx_str),rx2  ; но если строка закончилась (слишком
длинная)
        dec   r0                  ; то топчемся на месте
rx2:  cjne  a,#0dh,rx4            ; если приняли 0d

; приняли конец строки (0d ) и переходим на её обработку
        mov   a,r0                ; выясняем длину принятой строки
        clr   c
        subb  a,#rx_str
        mov   rx_len,a            ; записываем длину
        mov   r0,#rx_str          ; вновь ставим указатель на начало строки
        clr   sym_was             ; вновь отсеиваем ведущие пробелы
        clr   es                  ; запрет tx/rx прерывания пока не уберём строку в
другой буфер
        jmp   to_main

rx4:  mov   @r0,a                 ; скидываем принятый байт в строку
        inc   r0                  ; проходим дальше по строке
to_main: poppsw
        pop   acc
        reti                     ; возвращаемся в осн. программу

tx_strbyte:                ;передаёт байт символами f0h='f0'
        swap  a
        anl   a,#00001111b
        call  to_str
        mov   sbuf,a
        jnb   ti,$
        clr   ti
        mov   a,r1
        anl   a,#00001111b
        call  to_str
        mov   sbuf,a
        jnb   ti,$
        clr   ti
        ret

to_str:  cjne  a,#0ah,ts1         ; преобразует байт в его символ 0fh='F'
        jmp   ts2
ts1:  jnc   ts2
        add   a,#30h              ; от 0 до 9

```

```

ret
ts2: add a,#37h ; от А до F
ret

```

```

; 9600 бит/с, 9600 KHz, smod = 1, TH1 = 250d.

```

```

; 1старт(0)/8данных/1стоп(1) УАПП режим 1

```

```

;----- xxxxxxxxxxxx -----

```

```

start:

```

```

; mov p2,#0
; mov p1,#0
; mov p0,#0

```

```

mov SP,#mystack
mov PSW, #00
mov TMOD, #20h ; режим 2 t/c1, C/T=0 th перегружает tl
mov TCON, #00
mov SCON, #040h; режим 1 uapp 8N1
mov PCON, #80h ; smod=1 - 80h
mov IP, #00
mov IE, #00
mov TL1, #250d
mov TH1, #250d
setb TR1 ; запуск таймера tc1
mov 08h,#rx_str ; устанавливаем указатель на начало rx_str
setb NCstrobe
clr MCreedy ; контроллер готов

```

```

setb ea ; разрешаем прерывания
setb ti

```

```

setb REN ; разрешаем приём

```

```

begin:mov r1,#rx_str ; очищаем приёмную строку и буфер

```

```

mov r0,#rx_buf

```

```

b1: mov @r1,#0

```

```

mov @r0,#0

```

```

inc r0

```

```

inc r1

```

```

cjne r1,(rx_str+max_string+1),b1

```

```

setb es ; разрешаем приём/передачу

```

```

b2: clr main_lamp

```

; ждём приёма строки

jb es,b2

; приняли строку

mov a,rx_str

jz begin ; если принятая строка пуста (приняли 0d и всё) то мимо ушей

mov dptr,#ctable ; загружаем адрес таблицы опознаваемых команд

mov r1,#rx_str ; адрес опознаваемой строки

mov r2,#2 ; 2 символа на служебные поля (AJMP)

call search ; опознаём строчку по таблице

jb str_ok,v3

mov dptr,#rx_error ; если команду не отыскали то в dptr адрес куда перейти

; нашли совпадающую команду

v3: mov a,r1 ; вычисляем длину команды (чтобы вычислить длину данных)

clr c

subb a,#rx_str

clr c

mov r0,a ; вычисляем длину данных

mov a,rx_len

subb a,r0

mov rx_buf_len,a; запоминаем её в ячейке

mov r0,#rx_buf ; пересылаем строчку (кроме самой команды) в буфер для обработки соотв. программой

v6: mov a,@r1

mov @r0,a

inc r1

inc r0

cjne r1,#(rx_str+max_string+1),v6 ; пока не перешлём всю

; setb es ; разрешаем использование rx_str и прерывания

; переходим на выполнение принятой команды

mov a,#(low(return)) ; закидываем в стек адрес возврата

push acc

mov a,#(high(return))

push acc

clr a

setb main_lamp

jmp @a+dptr ; а там стоит AJMP, а в подпрограмме RET

; снова идём на приём команды

return: jmp begin

; поиск строки в таблице

search: ; ищем строчку R1 в таблице по dptr, в R2 длина служебных полей

```
    clr    str_ok
sr9:  cjne  @r1,#',sr8 ; убираем ведущие пробелы
      inc  r1
      jmp  sr9
```

```
sr8:  mov  a,r1
      mov  r0,a
      clr  a ; таблица кончилась ?
      movc a,@a+dptr
      jnz  sr1 ; да, выходим без флага ОК
      ret
```

```
sr1:  clr  a
      movc a,@a+dptr
      mov  03,@r0 ; из памяти в r2 (02)
      cjne a,03,sr2 ; символы сравнились (A и R2) ?
      jz   sr3 ; да, эти символы 0
```

```
      inc  r0
      inc  dptr
      jmp  sr1
```

```
sr2:  cjne  @r0,#',sr5 ; символы не сравнились => проверяем на пробел
      jnz  sr5
```

```
sr3:  setb  str_ok ; опознали команду (сравнился пробел и 0)
```

```
sr5:  clr  a ; эта ячейка не совпала ищем следующую
      movc a,@a+dptr ; ищем 0 в таблице
      jz   sr4
```

```
      inc  dptr
      jmp  sr5
```

```
sr4:  inc  dptr
      jb   str_ok,sr6 ; пропускаем служебные ячейки
      mov  a,r2
```

```
sr7:  jz   sr8 ; нашли следующую ячейку, идём сравнивать вновь
      inc  dptr
      dec  a
      jmp  sr7
```

; нашли совпадающую команду

```
sr6:  mov  a,r0
      mov  r1,a
      ret
```


ret

rx_error:

```
    mov    r0,#rx_buf
    call   tx_stringr
    mov    dptr,#command_err
    call   tx_string
    ret
```

help: nop

```
    mov    dptr,#help_str
    call   tx_string
    ret
```

load:

```
    mov    dptr,#load_str
    call   tx_string
    mov    r0,#rx_buf
    call   tx_stringr
    mov    a,#0dh
    call   say
    ret
```

status: mov a,#0ffh ; передаём в ПК текущий статус ЧПУ

```
    jnb    NCinput,st1
    clr     acc.0
```

```
st1: jnb    NCstart,st2
    clr     acc.1
```

```
st2: mov    sbuf,a
    ret
```

send:

```
    mov    r0,#rx_buf
```

```
se2: mov    a,@r0
    jnz     se1
    ret
```

```
se1: mov    p0,a
    clr     NCstrobe
    call    wait
    setb    NCstrobe
    call    wait
```

```

        inc    r0
        jmp    se2

smallsend:                ; принимаемая строка передаётся в ЧПУ
        jnb    ri,$        ; первый байт у строки - её длина
        clr    ri          ; 'ss'+$0d+length+N1...Nlength
        mov    a,sbuf
        mov    r0,a
        mov    r1,#rx_buf

        jz     ss3

ss1:    jnb    ri,$        ; принимаем данные в буфер
        clr    ri
        mov    @r1,sbuf
        inc    r1
        djnz   r0,ss1

        mov    r1,#rx_buf
ss2:    jnb    NCinput,ss5    ; если ввод исчез то хана
        mov    a,#'F'
        jmp    ss4
ss5:    jb     NCstart,ss2    ; старт можем и подождать
        mov    p0,@r1
        clr    NCstrobe
        call   wait
        setb   NCstrobe
        call   wait
        inc    r1
        djnz   acc,ss2
ss3:    mov    a,#'R'
ss4:    jnb    ti,$
        clr    ti
        mov    sbuf,a
        ret

end

```

Приложение В

Текст программы PERFOVW.PAS

```
uses dos,crt;
```

```
const
```

```
helptxt:array[1..5] of string[40] = (  
'Программа просмотра файлов перфолент',  
'формат вызова:   perfovw.exe file',  
'file -   имя просматриваемого файла',  
'           для справок:',  
'Сучков Д.И. Адаптация САПР PCAD ...'  
);
```

```
var
```

```
f      : file of byte;           { просматриваемый файл }  
pos    : longint;               { позиция в файле }  
key    : byte;                 { код нажатой клавиши }  
line   : byte;                 { количество линий перфоленты на экране }  
r      : registers;  
i      : byte;  
pst    : string;               { параметр файла }
```

```
function no_minus(a:integer):longint; { если аргумент отрицательный то }  
begin                                { функция равна нулю }  
if a<0 then no_minus:=0 else no_minus:=a;  
end;
```

```
function hex(d:byte):string;      { переводит число в строку с его hex символом }  
var a:byte;  
function h(d:byte):byte;  
begin  
if d<10 then h:=d+48 else h:=d+55  
end;  
begin  
a:= d shr 4 ;  
hex:=chr(h(a and $0f))+chr(h(d and $0f));  
end;
```

```
function viewable(a:char):char; { заменяет невидимые символы пробелом }  
begin
```



```

if a=#$0a then a:='-';           { символ конца кадра }
if ord(a)<32 then viewable:=#32 else viewable:=a;
end;

procedure perfoview;             { отображение образа перфоленты на экране }
var
a,y:byte;
s: string;
begin
if pos > filesize(f) then pos:=no_minus(filesize(f)-line);
seek(f,pos);
gotoxy(10,1);write(pos:5);       { печатаем текущую позицию в файле }
y:=1;                           { устанавливаем начальную позицию y }
while y<>line do begin          { отображаем line строчек }
read(f,a);                      { считываем очередной байт }
s:="";
for i:=7 downto 0 do            { формируем очередную строку }
if (a and trunc(exp(i*ln(2))))<>0 then s:=s+'*' else s:=s+'-';
insert(' ',s,6);
gotoxy(25,y+2);
write(s,' ',viewable(char(a and $7f)),' ',hex(a)); { печатаем линию перфоленты }
inc(y);                        { следующая строка }
end;
end;

begin
pst:=ParamStr(1);
{pst:='725034.pfl';}
if (pst='/?') or( pst='') then begin { заказали просмотр подсказки или не указали
файл ? }
writeln("");                   { да }
for i:=1 to 5 do begin
write(' ');
writeln(helptxt[i]);end;
exit;
end
else begin
r.ah:=1;r.ch:=$20;             { выкл. курсор }
intr($10,r);
line:=hi(windmax)-1;           { число отображаемых линий на экране }
clrscr;                        { очищаем экран }
gotoxy(1,1); write('Позиция:      76543 210 ----- Символ');
gotoxy(43,2); write('----- Байт');

assign(f,pst);                  { инициализируем файл }
{$i-}reset(f);{$i+}

```

```

if ioresult<>0 then begin      { такой файл не найден }
  write('Файл ',pst,' не найден ');
  exit; end;                  { => выходим }
repeat
  perfoview;                  { показываем образ с текущей позицией }
  key:=ord(readkey);          { вводим клавишу }
  case key of
    $48 : if pos<>0 then dec(pos);      { вверх }
    $50 : inc(pos);                      { вниз }
    $49 : pos:=no_minus(pos-line);      { pgup }
    $51 : inc(pos,line);                 { pgdown }
    $84 : pos:=0;                        { ctrl pgup }
    $76 : pos:=filesize(f)+1;            { ctrl pgdown }    { специально
переполняемся }
    $1b : exit;                          { esc }
  end;
until false;
end;
end.

```

Приложение Г
Текст программы C60TODRL.PAS

```
const
step  : integer = 4;           { константа для преобразования в дюймовую
систему }
start_str  : string = '/DBGRID 1'+#$0d+#$0a+'/DBUNIT 2'+#$0d+#$0a+'M72';

helptxt:array[1..5] of string[40] = (
'Программа конвертации файла в формате C60',
'системы КЛАРА в формат DRL PCAD ',
'формат вызова:  c60todrl.exe file',
'file - имя файла с C60 файлом',
);

var
f      : text;
ff     : text;
t      : array [0..9] of real;
oldx,oldy  : integer;
s,pst  : string;
namedrl  : string[12];
i       : byte;

function diameter:string;      { выдает строку с диаметрами инструментов }
var s:string;
function getone:real;           { считывает диаметр из строки }
var a:string; r:real; rr:integer;
begin
while s[1]=' ' do delete(s,1,1); { убираем ведущие пробелы }
if (pos(' ',s)=0) or (s='') then
begin getone:=0; exit; end;      { если все считали то =0 }
a:=copy(s,1,pos(' ',s)-1);      { a=текущему диаметру }
delete(s,1,pos(' ',s));         { убираем текущий диаметр из s }
val(a,r,rr);                    { преобразуем в число }
if rr=0 then getone:=r else getone:=0; { если ошибка то =0 }
end;

begin
repeat
repeat readln(f,s);
until (s[1]='#') or (eof(f));    { ищем символ # }
until (pos('Диаметры',s)<>0) or (eof(f));
delete(s,1,11);                 { убираем 'Диаметры' }
delete(s,ord(s[0])-1,2);        { убираем 0d из строки }
```

```
for i:=1 to 9 do t[i]:=getone;
end;
```

```
procedure getxy; { присваивает значения x,y исходя из
содержания строки S }
var p,i : byte;
k,x,y : longint;
function tonum(s:string;a:char):longint; { выдает численное значение метки A }
var r,rr:integer;
begin
p:=pos(a,s);
delete(s,1,p);
s:=copy(s,1,pos(' ',s)-1);
val(s,r,rr);
if rr=0 then tonum:=r else tonum:=0;
end;
procedure xyout; { выводим X,Y учитывая систему единиц }
function form(a:longint):string; { добавляет ведущие нули }
var s,c:string;
begin
c:='000000';
str(a,s);
delete(c,1,ord(s[0]));
form:=c+s;
end;
begin
writeln(ff,'X',form(x*step),'Y',form(y*step));
end;
begin
readln(f,s); { считываем строку }
delete(s,1,1); { убираем символ N }
while (s[1]<='9')and(ord(s[0])<>0) do delete(s,1,1); { убираем номер кадра }
i:=1;
while i<=ord(s[0]) do { вставляем пробелы между командами }
if s[i]>'9' then begin insert(' ',s,i); inc(i,2); end else inc(i);
s:=s+' '; { вставляем заключительный пробел }
if pos('T',s)<>0 then begin
k:=tonum(s,'T');
if k=0 then begin { диаметр =0, как признак конца текста }
readln(f,s);readln(f,s); writeln(ff,'M30'); s:=""; end
else writeln(ff,'T',k,'C',t[k]:3:2);
end;
if (pos('X',s)<>0) or (pos('Y',s)<>0) then begin
if pos('X',s)<>0 then x:=tonum(s,'X') else x:=oldx;
if pos('Y',s)<>0 then y:=tonum(s,'Y') else y:=oldy;
xyout;
```

```

if pos('J',s)<>0 then begin for i:=1 to tonum(s,'H') do begin      { если цикл отверстий
по y }
  inc(y,tonum(s,'J')); xyout; end;
end;
if pos('I',s)<>0 then begin for i:=1 to tonum(s,'H') do begin      { если цикл отверстий
по x }
  inc(x,tonum(s,'I')); xyout; end;
end;
oldx:=x; oldy:=y;          { запоминаем x,y }
end;
end;

begin
pst:=ParamStr(1);
{pst:='test.c60';}
if (pst='/?') or( pst='') then begin          { заказали просмотр подсказки или не указали
файл ? }
  writeln("");
  for i:=1 to 5 do begin
    write(' ');
    writeln(helptxt[i]);end;
  end
else begin
  assign(f,pst);                          { инициализируем файл .c60 }
  {$i-} reset(f); {$i+}
  if ioresult<>0 then begin
    write('Файл ',pst,' не найден ');exit; end;  { такой файл не найден }
    namedrl:=copy(pst,1,pos('.',pst+'.')-1)+'.drl';
    {$i-} assign(ff,namedrl); erase(ff); {$i+}    { уничтожаем старый drl файл }
    if ioresult=0 then;
    assign(ff,namedrl); rewrite(ff);          { создаем новый drl файл }

    writeln(ff,start_str);                    { пишем заголовок файла }
    diameter;                                { считываем таблицу диаметров }
    repeat readln(f,s); until s[1]='%';      { ищем начало программы ЧПУ }
    while not eof(f) do begin
      getxy;
      end;
      flush(ff);                             { сбрасываем буфер на диск }

    end;
  end.

```

Приложение Г
Текст программы PFLOAD.PAS

uses crt;

{

Программа предназначена для считывания перфоленты при помощи перфосчитывателя CONSUL и записи образа перфоленты на диск компьютера. При этом в отдельном файле сохраняется название файла с образом и комментарий к нему

28 апреля 1998 г.

Сараев К.В.5:5005/14.34

}

label newtape,reload;

const

nametmp : string = 'PF';

maxnum : longint = 999999; { максимальный номер для нумерации
файлов }

perfoext : string[3] = 'pfl'; { расширение для файлов с образом перфоленты }

namemask : string = '000000';

descriptfile : string = 'file_id.diz'; { имя файла с описаниями к перфолентам }

var

f : file of byte;

des : text; { переменная для файла с описаниями }

s,pst : string;

comment : string; { комментарий к текущей ленте записываемый
в файл с описанием }

i : longint;

ba : word; { адрес порта }

prttmp : byte; { времянка для сохранения содержимого порта
управления }

data : byte;

choice : byte;

counter : longint; { счетчик принятых байт }

procedure error(s:string); { выводит сообщение об ошибке s и выходит из
программы }

begin

write(s);

halt;

end;

```

function status(n:byte):boolean;      { выдает состояние n бита статуса пар. порта }
var a:byte;
begin
  {n:=trunc(exp(n*ln(2)));           { возводим 2 в степень n }
  a:=port[ba+1];
  status:=boolean(n and a);
end;

```

```

function strobe_in:boolean;
begin
  strobe_in:=not status($80);      { подключено к линии /busy }
end;

```

```

function ready_in:boolean;
begin
  ready_in:=status($10);          { подключено к линии /slct }
end;

```

```

procedure start(a:boolean);      { выставляет старт если true или убирает его }
begin
  prttmp:=((prttmp and ($ff and (not 8)) ) or (8*byte(not a)) );    { slctin }
  port[ba+2]:=prttmp;
end;

```

```

procedure toforward(a:boolean);  { выставляет прямое движение или обратное }
begin
  prttmp:=((prttmp and ($ff and (not 1)) ) or (1*byte(a)) ); { strobe }
  port[ba+2]:=prttmp;
end;

```

```

procedure statusvw;              { показывает текущее состояние входных линий }
begin
  gotoxy(3,20);if strobe_in then highvideo else lowvideo; write('strobe');
  gotoxy(3,21);if ready_in then highvideo else lowvideo; write('ready');
end;

```

```

procedure txttemplate(a,b:string);
begin
  gotoxy(2,2);write('Текущий файл: ',a);
  gotoxy(2,4);write('Комментарий: ',b);
end;

```

```

begin
  ba:= memw[$40:$08];            { узнаем адрес порта LPT1 }

```

```

if ba=0 then error('Отсутствует адаптер параллельной связи LPT1');
prtmp:=$20; { для режима считывания из LPT (EPP mode) }
port[ba]:=$ff;
port[ba+2]:=$20; { включаем режим считывания с порта }
start(false); { выключаем движение ленты }
toforward(true); { прямое направление движения ленты }
assign(des,descriptfile);
{$i-}append(des);{$i+} { пытаемся продлить файл с описаниями }
if ioresult<>0 then rewrite(des); { если он не существует то создаем }
append(des);

```

```

newtape:
clrscr;
comment:="";
for i:=0 to maxnum do begin { находим свободное имя файла }
  str(i,s);
  s:=copy(namemask,1,6-length(s))+s; { заменяем в имени допустим 1 на 000001 }
  pst:=nametmp+s+'.'+perfoext;
  assign(f,pst); { инициализируем файл }
  {$i-}reset(f);{$i+}
  if ioresult<>0 then i:=maxnum; { нашли незанятое имя файла }
  {$i-}close(f);{$i+}; if ioresult<>0 then;
end;
gotoxy(2,2);write('Текущий файл: ',pst);
gotoxy(2,4);write('Введите комментарий к этой ленте: ');
readln(comment);
writeln(des,pst+#09+comment); flush(des); { записываем в файл с описаниями }

```

```

reload:
counter:=0;
clrscr; txttemplate(pst,comment);
gotoxy(8,7);write('Вставьте ленту и подготовьте перфосчитыватель ');
gotoxy(8,8);write('Перфосчитыватель не готов !');

```

```

repeat until ready_in; { ждем когда подготовят перфосчитыватель }

```

```

rewrite(f); { создаем/перезаписываем текущий файл }
gotoxy(8,8);write(' ');

```

```

start(true); { включаем протяжку ленты }
repeat
  repeat { ждем восходящий фронт у строба }
  until (status($80))or(not status($10) );
  repeat until not status($80) or (not status($10) );
  if status($10) then begin

```



```

data:=port[ba];
write(f,data);
inc(counter);
gotoxy(2,10);write('Считано: ',counter,' байт');
end;
until not status($10);          { читаем до исчезания готовности перфосчитывателя
}

start(false);                   { убираем старт }

gotoxy(2,10);write('Считывание остановлено');
gotoxy(2,11);write('1 - заново считать эту ленту');
gotoxy(2,12);write('2 - считывание этой ленты закончено');
gotoxy(2,13);write('3 - выйти из программы');
repeat
gotoxy(2,15);write('Ваш выбор:      ');
gotoxy(13,15);readln(choice);
until (choice=1)or(choice=2)or(choice=3); { ждем правильный выбор }
close(f);
case choice of
1: begin erase(f); goto reload; end;
2: begin goto newtape; end;
3: halt;
end;

end.

```