

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Оренбургский государственный университет»

Кафедра вычислительной техники

Т.З. Аралбаев, Р.Р. Галимов, Р.И. Хасанов

ПРОЕКТИРОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Рекомендовано к изданию Редакционно-издательским советом
федерального государственного бюджетного образовательного учреждения
высшего профессионального образования
«Оренбургский государственный университет»
в качестве методических указаний для студентов, обучающихся по программам
высшего профессионального образования по направлению подготовки
230100.62 Информатика и вычислительная техника

Оренбург
2012

УДК 681.3(075)
ББК 32.973.2-02я7
А 79

Рецензент- кандидат технических наук, доцент А.В. Хлуденев

Аралбаев, Т.З.
А 79 Проектирование вычислительных систем: методические указания к лабораторным работам / Т.З. Аралбаев, Р.Р. Галимов, Р.И. Хасанов; Оренбургский гос. ун-т. - Оренбург: ОГУ, 2012. – 64 с.

Лабораторный курс содержит 9 лабораторных работ и методические указания к ним. Каждая работа включает теоретическое изложение материала, постановку задачи, порядок выполнения и контрольные вопросы для самоподготовки.

Лабораторный курс рекомендован преподавателям как вспомогательный материал в организации и проведении занятий, а также студентам по профилю подготовки - «Вычислительные машины, комплексы, системы и сети» - для аудиторного и самостоятельного освоения лабораторного курса дисциплины «Проектирование вычислительных систем».

УДК 681.3(075)

ББК 32.973.2-02я7

© Аралбаев Т.З., Галимов Р.Р.,
Хасанов Р.И., 2012
© ОГУ, 2012

Содержание

Введение.....	4
Обозначения и сокращения.....	5
1 Лабораторная работа № 1. Исследование целевой функции построения систем и выбор критерия для оценки эффективности проекта.....	6
2 Лабораторная работа № 2. Моделирование системы принятия решений в вычислительной системе	11
3 Лабораторная работа № 3. Определение вклада фактора в общий результат на основе принципа Парето	15
4 Лабораторная работа № 4. Прогнозирование временных рядов на основе уравнений регрессии	22
5 Лабораторная работа № 5. Оптимизация топологии распределенной вычислительной системы.....	28
6 Лабораторная работа № 6. Разработка программного обеспечения для микроконтроллера ADuC812.....	35
7 Лабораторная работа № 7. Разработка подсистемы ввода-вывода и регистрации сигналов в режиме реального времени.....	45
8 Лабораторная работа № 8. Методы защиты информации. Шифр Цезаря	52
9 Лабораторная работа № 9. Автоматизированное проектирование вычислительной системы с использованием инструментальных средств класса SCADA - «Trace Mode 6.0».....	56
10 Литература, рекомендуемая для изучения тем	63
Приложение А. Листинг программы.....	64

Обозначения и сокращения

CS - Communication System, коммуникационная система (каналы связи)

MTU - Master Terminal Unit, диспетчерский пункт управления, осуществляющий контроль и управление высокого уровня

RTU - Remote Terminal Unit, удаленный терминал, осуществляющий первоочередные задачи управления в режиме реального времени

SCADA - Supervisory Control And Data Acquisition, Диспетчерское управление и сбор данных

ВПД – внешняя память данных

ВПП – внешняя память программ

ВР – временной ряд

ВС – вычислительная система

ГЭ – графический элемент

ЖК – жидкокристаллический

ИД – источник данных

МК – микроконтроллер

МНК – метод наименьших квадратов

НСД – несанкционированный доступа

ОЗУ – оперативно запоминающее устройство

ПЛИС – программируемая логическая интегральная схема

РСКУ – распределенная система контроля и управления

СКД – система контроля доступа

СОД – средство обработки данных

ЦФ – целевая функция

Введение

Настоящий лабораторный практикум предназначен для получения практических навыков студентами специальности и профиля подготовки – «Вычислительные машины, комплексы, системы и сети» при изучении дисциплины «Проектирование вычислительных систем» (ПВС).

Лабораторный курс содержит девять работ, рассчитанных на 36 часов аудиторных занятий. Предлагаемые задания охватывают основные разделы рабочей программы, связанные с изучением технологии и организации проектирования вычислительных систем.

Общие методические рекомендации по использованию лабораторных работ и методических указаний:

- к выполнению лабораторной работы следует приступать после ознакомления с теоретической частью соответствующего раздела и рекомендациями, приведенными в конкретной работе;

- лабораторные работы рекомендуется выполнять в порядке их нумерации в аудиторное время, указанное в описании работы;

- рекомендуется для экономии времени отчеты о лабораторных работах оформлять в виде протоколов работы с обязательным указанием даты, номера, темы, цели работы и выводов с краткой характеристикой результата;

- дополнительные сведения по лабораторным работам содержатся в прилагаемом списке литературы.

Лабораторный курс может быть освоен на индивидуальном компьютере со средними техническими характеристиками. Обязательным для полной реализации курса является наличие комплекта офисных приложений и пакет прикладных программ MathCad, желательно иметь выход в Интернет.

Практикум рекомендован преподавателям как вспомогательный материал в организации и проведении занятий, а также студентам - для аудиторного и самостоятельного освоения лабораторной части дисциплины ПВС.

1 Лабораторная работа № 1. Анализ целевой функции и выбор критерия оценки эффективности построения вычислительных систем

Цель работы: выполнить исследование целевой функции построения систем и осуществить выбор критерия для оценки эффективности

1.1 Теоретическая часть

Целевая функция (ЦФ) – функция, характеризующая оценку эффективности разрабатываемой системы в зависимости от ее основных технико-экономических параметров. Обычно для выбора параметров ЦФ рассматривают ее изменение при некоторых ограничениях, например на стоимость разработки системы, или на ее производительность.

Параметры системы, определяющие экстремум (*min* или *max* ЦФ) функции, соответствуют оптимальным проектным параметрам.

Критерий оценки эффективности (качества) системы – некоторый параметр системы, входящий в целевую функцию, позволяющий производить сопоставление различных вариантов создаваемых систем.

Свойства критерия:

- наглядность;
- вычисляемость;
- простота понимания и получения данных для вычисления;
- непосредственная связь с целевой функцией.

Критерием может быть технический или экономический параметр системы. Обычно критерием выбирают один параметр. Если для сопоставления систем нужно использовать несколько критериев, то:

- производят «свертку» нескольких критериев в один обобщенный, т.е. определяют некоторое аналитическое выражение, содержащее в себе совокупность нескольких параметров оценки системы;
- определяют совокупность значений основных параметров системы, так назы-

ваемую область Парето, удовлетворяющую заданию на разработку.

В частности, целевая функция при разработке вычислительных средств контроля доступа в АСУ ТП должна учитывать минимизацию ущерба от несанкционированного доступа (НСД) при заданных стоимостных ограничениях на средства контроля.

В общем случае задачу защиты доступа можно свести к задаче распознавания личности, результатом решения которой является определение принадлежности идентифицируемого образа Q^x к одному из двух классов, в частности, классу A , представители которого имеют доступ к АСУ ТП, или классу B , представляющему категорию лиц, не имеющих права доступа. В зависимости от исхода распознавания принимается решение о допуске субъекта к информационному каналу АСУ ТП или о принятии соответствующих мер по отношению к лицу, отнесенному к категории B . В обоих случаях решение сопряжено с соответствующими ошибками распознавания, оцениваемыми вероятностью α при допуске к АСУ ТП образа из класса B , и вероятностью β – при отказе в допуске субъекту из класса A . В соответствии с результатами решения эффект E от идентификации одного субъекта может быть оценен согласно следующему выражению:

$$E = [e_A \cdot (1 - \alpha) + z_A \cdot \alpha] \cdot P_A + [e_B \cdot (1 - \beta) + z_B \cdot \beta] \cdot P_B, \quad (1.1)$$

где: e_A и e_B – возможные издержки на процедуру контроля при безошибочной идентификации субъекта, величины e_A и e_B могут отличаться друг от друга и соответствовать стратегии системы контроля доступа (СКД) при обнаружении образов из классов A и B ;

z_A и z_B – затраты от ошибочной идентификации субъекта и принятых мер при неверной стратегии СКД;

α и β – соответственно ошибки первого и второго рода при распознавании субъекта;

P_A и P_B – соответственно априорные вероятности появления на входе средств контроля доступа образов из класса A и B .

Эффект от использования контроля доступа в течение времени T при интенсивности заявок на доступ λ определяется по следующему выражению:

$$E = \{ [e_A \cdot (1 - \alpha) + z_A \cdot \alpha] \cdot P_A + [e_B \cdot (1 - \beta) + z_B \cdot \beta] \cdot P_B \} \cdot T \cdot \lambda. \quad (1.2)$$

Оценки P_A , P_B на некотором интервале T являются прогнозными, их величины зависят от интенсивностей потоков заявок, поступающих на идентификацию. Значения вероятностей P_A и P_B определяются из следующих соотношений:

$$P_A = \frac{\lambda_c}{\lambda_c + \lambda_{НСД}}, \quad P_B = \frac{\lambda_{НСД}}{\lambda_c + \lambda_{НСД}}, \quad \lambda = \lambda_c + \lambda_{НСД}, \quad (1.3)$$

где λ_c и $\lambda_{НСД}$ – соответственно интенсивности санкционированных и несанкционированных заявок на доступ.

В основе затрат z_B лежит материальный ущерб предприятию u , нанесенный в результате разового НСД. Размер u зависит от конкретного производства и определяется как средняя статистическая величина по имеющейся выборке издержек производства от НСД за интервал исследования работы предприятия.

Как видно из вышеизложенного, эффективность системы контроля доступа АСУ ТП определяется параметрами e_A , e_B , z_A , z_B , α , β , P_A , P_B . При построении системы необходимо подобрать значения параметров, которые обеспечивали бы минимальное значение целевой функции.

1.2 Порядок выполнения работы

1. Войти в среду MathCad. Создать новый документ;
2. Провести инициализацию переменных e_A , e_B , α , β , z_A , z_B , P_A , P_B , λ согласно варианту. Пример инициализации переменных представлен на рисунке 1.1;
3. Построить графики исследований ЦФ. Добавляем новый элемент «График функции», инициализируем переменные по оси ординат и абсцисс так, как показано на рисунке 1.2;


```

eA := 60      eB := 600      a := 0.85..0.95      b := 0.85..0.95      zA := 120      zB := 3000      PA := 0.77
PB := 0.64      T := 10000      L := 3000      i := 0..4

aMass := ⎛ 0.85 ⎞
          ⎜ 0.87 ⎟
          ⎜ 0.89 ⎟
          ⎜ 0.91 ⎟
          ⎜ 0.93 ⎟
          ⎜ 0.95 ⎟

bMass := ⎛ 0.85 ⎞
          ⎜ 0.87 ⎟
          ⎜ 0.89 ⎟
          ⎜ 0.91 ⎟
          ⎜ 0.93 ⎟
          ⎜ 0.95 ⎟

Ea_i := [eA · (1 - aMass_i) + zA · aMass_i] · PA + [eB · (1 - bMass_i) + zB · bMass_i] · PB
Eb_i := [eA · (1 - aMass_i) + zA · aMass_i] · PA + [eB · (1 - bMass_1) + zB · bMass_1] · PB

bUgol := tan(20/0.02)      aUgol := tan(0.3/0.02)
bUgol = 1.47      aUgol = -0.134

```

Рисунок 1.1 – Листинг программы для оценки эффективности проектируемой системы

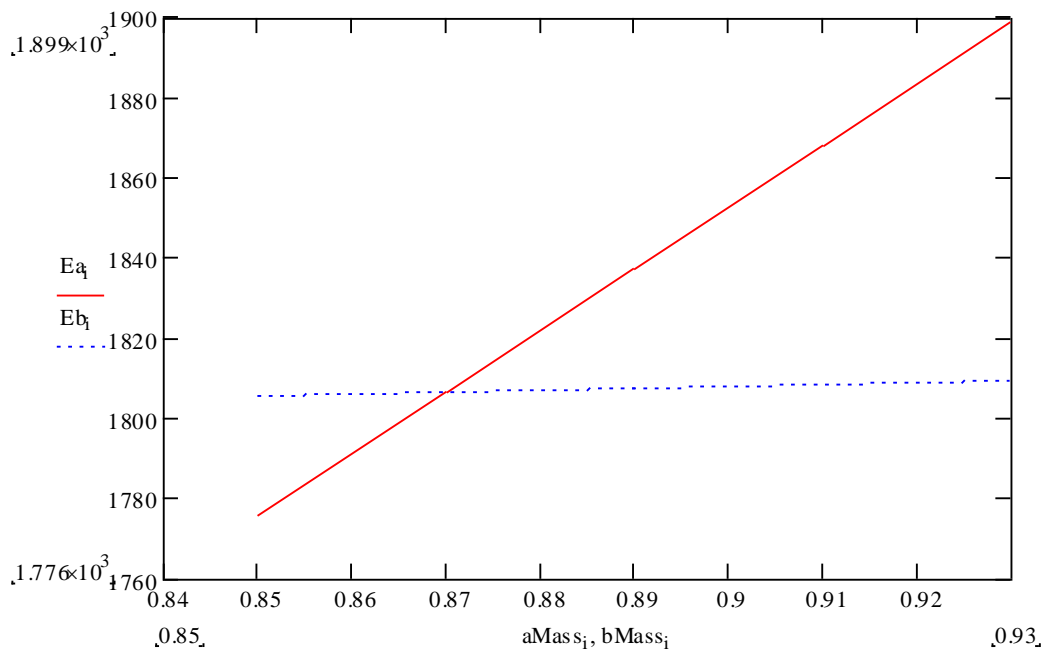


Рисунок 1.2 – График исследований целевой функции

4. Величины z_A и z_B определяются типом АСУ ТП и связаны с возможными видами ущерба, специфичного для конкретного производства, причем z_A также, как α , и β характеризуют технико-экономические параметры системы контроля доступа и могут варьироваться при разработке системы. На этом основании можно считать досто-

верность контроля, определяемую параметрами α , и β , основным критерием оценки качества и выбора системы контроля и управления доступом. При этом к факторам ограничительного характера относятся стоимостные затраты на создание системы и время идентификации пользователя.

1.3 Задание на лабораторную работу

1. Провести исследование заданной целевой функции для оценки эффективности системы в соответствии с вариантом задания и выбрать критерий оценки.
2. Просчитать параметр E при изменении параметров α и β в заданном диапазоне с шагом 0.02. При вычислениях один из параметров необходимо фиксировать.
3. Определить, изменение какого параметра больше влияет на величину ЦФ.
4. Выбрать критерий для оценки системы контроля доступа.

1.4 Варианты заданий

Варианты заданий для выполнения лабораторной работы представлены в таблице 1.1. Параметр $T = 1000$ для всех вариантов.

Таблица 1.1 – Варианты заданий для выполнения лабораторной работы

Вариант	eA	eB	α	B	zA	zB	PA	PB	Λ
1	100	1000	0,85-0,95	0,85-0,95	200	5000	0,95	0,98	3000
2	90	900	0,85-0,95	0,85-0,95	180	4500	0,9	0,88	3000
3	80	800	0,85-0,95	0,85-0,95	160	4000	0,86	0,79	3000
4	70	700	0,85-0,95	0,85-0,95	140	3500	0,81	0,71	3000
5	60	600	0,85-0,95	0,85-0,95	120	3000	0,77	0,64	3000
6	50	500	0,85-0,95	0,85-0,95	100	2500	0,74	0,58	3000
7	40	400	0,85-0,95	0,85-0,95	80	2000	0,7	0,52	3000
8	30	300	0,85-0,95	0,85-0,95	60	1500	0,66	0,47	3000
9	20	200	0,85-0,95	0,85-0,95	40	1000	0,63	0,42	3000
10	10	100	0,85-0,95	0,85-0,95	20	500	0,6	0,38	3000

1.5 Контрольные вопросы

1. Каким образом производится расчет ошибок первого и второго рода.
2. Что понимается под оптимизацией проектных решений.
3. Привести примеры целевых функций, используемых при проектировании

вычислительных систем.

4. Привести примеры критериев экономического, технического и социального вида для оценки качества проектируемой системы.

5. Может ли значение целевой функции быть критерием оценки качества проекта.

2 Лабораторная работа № 2. Моделирование системы принятия решений в вычислительной системе

Цель работы: построить модель и исследовать процесс принятия решения с использованием ВС

2.1 Теоретическая часть

Одной из моделей принятия решений является модель игры с природой, позволяющая пользователю по принципу минимакса (или другого правила) выбрать стратегию поведения ВС в ответ на случайно заданную стратегию природы.

Эта модель может быть использована при выборе стратегии защиты информации в ответ на случайные угрозы информационной безопасности ВС.

Для противодействия стратегиям природы согласно принципу Эшби мощность множества возможных стратегий ВС $Q2$ должна быть не меньше мощности возможных стратегий природы $Q1$, т.е.:

$$Q2 \geq Q1. \quad (2.1)$$

Изучение условия задачи о выборе стратегии ВС позволяет свести ее к задаче из теории игр с природой, в которой природа случайным образом определяет характеристики несанкционированного доступа в интервале T , а подсистема принятия решений ВС в соответствии со стратегией природы выбирает собственную оптимальную стратегию защиты. Модель выбора оптимальной стратегии определена в виде матрицы исходов стратегий размером $n \times m$, представленной в таблице 2.1.

Таблица 2.1 – Матрица исходов стратегий.

S C	s_1	s_2	...	s_m	E
c_1	ef_{11}	ef_{12}	...	ef_{1m}	E_1
c_2	ef_{21}	ef_{22}	...	ef_{2m}	E_2
...
c_n	ef_{n1}	ef_{n2}	...	ef_{nm}	E_n

Строки матрицы соответствуют стратегиям защиты BC , а столбцы – прогнозируемым стратегиям природы на перспективу T . Каждый элемент матрицы содержит оценки параметра эффекта ef_{nm} , соответствующие экономическому (или другому эффекту), для конкретной стратегии, представленные в виде дроби e_{ij}/z_{ij} , числитель которой характеризует сэкономленное качество от правильно выбранной стратегии защиты, а знаменатель – затраты от применения i -ой стратегии при ликвидации j -ой угрозы ($i=1-n, j=1-m$). Выбор стратегии c_i из множества C производится с учетом условия каждой задачи, например - по максимуму экономии или минимуму затрат с учетом следующего выражения:

$$c \in C : \max | \min \sum_{j=1}^m ef_{ij}, \quad i = 1, n; \quad j = 1, m. \quad (2.2)$$

Эффект от каждой стратегии, в случае ее использования, определяется экспертами и закладывается в матрицу C . В случае несоответствия стратегий множества C требуемой величине эффекта ставится вопрос о поиске и разработке новых стратегий BC .

В условиях риска выбор стратегии BC часто производят на основе критериев минимакса или максимина. В данной работе под критерием минимакса понимается выбор максимальной экономии из минимальных значений по каждой страт вычислительной системы. Соответственно, критерий максимина рассматривается как минимальное значение максимальных затрат по каждой стратегии BC .

2.2 Задание на лабораторную работу

1. Построить программную модель матрицы всех возможных исходов стратегий по виду таблицы матрицы исходов стратегий. Размер матрицы соответствует $R=N+10$, где N – порядковый номер студента по журналу;

1. С использованием генератора случайных чисел (ГСЧ) заполнить матрицу значениями экономии от применения стратегии ВС e_{ij} (для нечетных вариантов) и затрат z_{ij} (для четных вариантов). Диапазон изменения значений 1-10;

2. С использованием ГСЧ выбрать два номера наиболее вероятных случайных стратегий природы из диапазона $1-R$, например 5 и 10 (в общем случае число наиболее вероятных стратегий может быть любым из диапазона $1-R$);

3. Для данных стратегий природы по принципу минимакса для нечетных вариантов и максимина – для четных вариантов выбрать оптимальную стратегию ВС;

4. Сравнить эффективность выбранной стратегии ВС с другими стратегиями ВС для конкретной стратегии природы. Для этого распечатать таблицу;

5. Сделать выводы по работе.

2.3 Порядок выполнения работы

1. Войдите в среду Excel. Создайте новый документ, перейдите на первый лист этого документа. Заполните таблицу матрицы исходов стратегий случайными числами, как показано на рисунке 2.1. Размер матрицы соответствует $R=N+10$, где N – порядковый номер студента по журналу. В рассматриваемом примере R равно 15;

2. Выберите 2 стратегии природы, которые наиболее вероятны, например, 3 и 9. Для реализации стратегии максимина в ячейке $P1$ (в столбце следующем за матрицей исходов) запишите формулу «=**МАКС**($C1;J1$)» (для стратегии минимакса выберите функцию «**МИН**»). Функция «**МАКС**» возвращает максимальное значение из списка аргументов (функция «**МИН**» возвращает минимальное значение из списка аргументов);

3. В ячейках $P2-P15$ аналогично запишите формулу «=**МАКС**($Cx;Jx$)», где x – изменяет свои значения от 2 до 15 для соответствующих ячеек;

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	c\c	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Max
2	1	8,515794	6,314989	6,085425	6,912203	6,74166	9,657677	0,173887	2,518103	9,20689	8,855673	7,783316	3,506682	2,753653	3,793635	1,086138	8,855673
3	2	4,450796	9,450801	6,807543	4,546047	7,303199	3,891633	3,718511	2,137972	1,478172	5,413421	8,918054	5,185484	9,203729	8,557619	3,022995	6,807543
4	3	3,338058	3,072899	3,862867	8,926565	8,837181	1,926437	4,697968	6,805138	1,320536	3,30051	7,374337	8,843751	8,002101	5,327553	3,670887	3,862867
5	4	1,12131	8,183262	7,604448	6,479868	4,278717	1,313011	8,242451	8,02521	6,816758	1,764848	8,090678	4,657665	0,225172	4,052095	8,828196	7,604448
6	5	6,140192	8,891859	2,38072	1,116998	6,018871	0,157015	2,063777	7,918194	5,772063	7,595655	7,657386	0,977584	5,590813	8,760308	0,369953	7,595655
7	6	4,935811	9,446705	6,912048	4,725637	0,997561	9,367694	0,867769	9,883139	9,03716	3,389664	8,197019	2,686939	4,213152	5,946193	8,313076	6,912048
8	7	7,838042	7,507299	5,915197	5,741888	2,567749	0,150598	6,112698	0,905062	0,810593	4,057633	9,899912	3,571845	9,101491	0,670921	6,63396	5,915197
9	8	3,46634	8,420075	8,8378	5,031767	9,565976	8,179015	7,719799	6,967299	4,493681	1,496538	0,291867	3,945647	4,616263	4,792456	0,495684	8,8378
10	9	8,540318	4,052918	8,860361	0,440798	1,023764	2,671314	3,955807	8,957137	8,338114	8,927289	0,641513	5,683205	4,661737	6,73107	3,336639	8,927289
11	10	0,616889	2,776301	1,579424	8,267636	0,316832	4,557573	0,546292	1,794942	1,629519	0,7812	8,890356	2,39218	7,71519	7,264638	8,947567	1,579424
12	11	8,7753	4,856308	5,863175	4,38622	7,569577	4,237352	6,797613	3,406498	3,484086	0,935659	0,526879	1,217752	3,358072	6,519284	4,821688	5,863175
13	12	1,474963	3,299255	8,083438	9,433226	1,252622	5,241158	6,764074	2,701371	5,673054	7,503026	9,852066	1,539217	8,241164	6,728447	3,122985	8,083438
14	13	9,675559	5,402553	9,816329	5,462374	7,217485	2,070884	8,818816	9,784753	3,003496	0,498651	6,796574	9,661187	0,936755	1,128358	1,79494	9,816329
15	14	8,17609	5,423198	6,953397	5,789779	0,507347	7,361365	4,830256	0,084985	0,293323	0,475354	4,14308	7,209742	8,729187	5,785058	7,78969	6,953397
16	15	0,414771	5,28969	7,906192	1,905351	2,716275	2,866687	0,098462	8,851987	3,547055	7,749165	5,638656	4,309318	5,572921	6,765193	9,001971	7,906192
17																min=	1,579424

Рисунок 2.1 – Пример матрицы исходов стратегий

4. В ячейке **P16** запишите формулу «=**МИН(P1:P15)**»;

5. Исследуйте полученный процесс принятия решения самостоятельно.

В случае максимина по каждой стратегии пользователя сначала определяют максимальные затраты от стратегии ВС на прогнозируемые стратегии природы и выписывают их в последней колонке. Затем выбор стратегии осуществляется по принципу минимальных затрат из максимально возможных.

Во втором случае по каждой стратегии пользователя на прогнозируемые стратегии природы сначала определяют минимальные экономии от стратегии и выписывают их в последней колонке. Затем выбор стратегии осуществляется по принципу максимальной экономии из минимально возможных вариантов.

2.4 Контрольные вопросы

1. Изложить принцип Эшби.
2. Объясните стратегии минимакса и максимина.
3. Привести примеры программных средств, которые позволяют моделировать системы принятия решений.
4. Привести примеры аппаратно-программных средств, которые позволяют моделировать системы принятия решений.

3 Лабораторная работа № 3. Определение вклада фактора в общий результат на основе диаграммы Парето

Цель работы: освоить технологию расчета и построения диаграммы Парето в среде MS Excel

3.1 Теоретическая часть

Анализ Парето – это способ исследования и организации данных, предложенный в 1897 году итальянским экономистом Вильфредо Парето. Анализ Парето может быть применен для решения различных экономических и технических задач, например, для распределения финансовых средств, для принятия мер при оптимизации ситуации. На основе анализа строится диаграмма Парето, позволяющая выделить «жизненно важное меньшинство» по сравнению с «незначительно важным большинством». В результате анализа был сформулирован так называемый «принцип Парето» или принцип соотношения «20-80», который подтверждается количественными исследованиями в самых различных сферах жизни. Так, 20% товаров определяют 80% доходов компании; 20% преступников совершают 80% преступлений; 20% имеющейся одежды люди носят в течение 80% времени. Можно найти немало полезных идей в результате исследований с применением этого принципа. Например, можно оценить долю действительно нужных книг в шкафу, долю полезной информации в газете, долю нужных файлов на диске компьютера.

Основной задачей настоящей работы является освоение технологии быстрого построения диаграммы Парето с использованием средств автоматизации приложения MS Excel. Построение диаграммы Парето рассмотрим на примере некоторой обобщенной задачи выявления перечня факторов и вкладов (долей) факторов в общий результат. Факторами могут быть перечень мер для повышения производительности труда, перечень заболеваний и ущерб от каждого заболевания для предприятия, перечень правонарушений и затраты на борьбу с этими правонарушениями.

3.2 Постановка задачи

1. Ознакомиться с назначением и порядком работы с MS Excel для построения диаграммы Парето в соответствии с заданием по варианту.
2. В среде MS Excel составить таблицу исходных и расчетных данных задачи.
3. На основе диаграммы Парето определить вклады факторов в общий результат.
4. Составить отчет исследований по пунктам 2-3.
5. Ответить на контрольные вопросы для самопроверки.

3.3 Порядок выполнения работы

1. В среде Excel составить таблицу 3.1 исходных и расчетных данных задачи. Вклады факторов в общий результат во втором столбце необходимо расположить в порядке их убывания с помощью команды «Сортировка - По убыванию».

2. В первом столбце указать наименование соответствующего фактора. Для построения диаграммы Парето необходимо рассчитать в процентах долю вклада каждого фактора от общей суммы вкладов (столбец 3) и данные Парето (столбец 4). Данные Парето получены постепенным накапливанием долей каждого фактора. Первая строка столбца 4 совпадает со значением первой строки столбца 3. Вторая строка столбца 4 получена суммированием значения первой строки столбца 4 и значения второй строки столбца 3, $(35.77 + 26.69 = 62.46)$. Третья строка столбца 4 получена суммированием значения второй строки столбца 4 и значения третьей строки столбца 3 $(62.46 + 14.10 = 76.55)$ и т.д., до получения последнего значения столбца 4. О корректности вычислений свидетельствует число 100 в строке последнего фактора, соответствующее 100% результата.

3. Для построения диаграммы Парето выделим данные первого, второго и четвертого столбцов (для выборочного выделения используют клавишу «Ctrl»), не выделяя сумму.

Таблица 3.1 – Таблица расчета диаграммы Парето

Факторы	Вклад фактора (ден.ед.)	Доля фактора (%)	Данные Парето (%)
Ф1	8768	35,77	35,77
Ф2	6543	26,69	62,46
Ф3	3456	14,10	76,55
Ф4	2341	9,55	86,10
Ф5	1231	5,02	91,12
Ф6	976	3,98	95,11
Ф7	834	3,40	98,51
Ф8	234	0,95	99,46
Ф9	132	0,54	100,00
Сумма	24515		

4. Для Excel 2003 необходимо в режиме «Мастера диаграмм» (или «Вставка → Диаграмма») выбрать тип диаграммы («Нестандартные → График гистограмма 2»), позволяющей отобразить трехосевую диаграмму (рисунок 3.1).

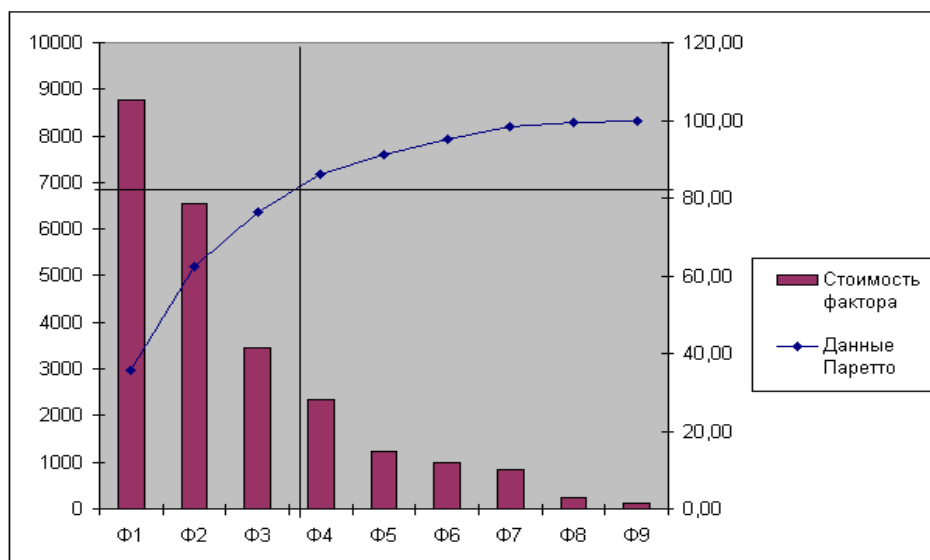


Рисунок 3.1 – Диаграмма Парето

Для Excel 2007 необходимо на вкладке «Вставка» выбрать команду «Гистограмма→Гистограмма с группировкой». Выделить ряд «Данные Паретто» и нажать правой кнопкой мыши выбрать команду «Формат ряда данных». В разделе «Параметры ряда» установить пункт «Построить ряд» в значение «По вспомогательной оси». Далее зарываем данное диалоговое окно.

Снова выделите ряд «Данные Паретто» и в контекстном меню выберите пункт

«Изменение типа диаграммы». В появившемся окне выберите тип диаграммы «График».

6. Проанализируем диаграмму. Проведем горизонтальную прямую, соответствующую 80% вкладов факторов до пересечения с графиком вкладов (в общем случае анализ проводится для любого числа процентов). Слева от точки пересечения размещены факторы, обеспечивающие 80% результата. На реальных данных можно проверить справедливость принципа Парето.

Для ряда задач построение диаграммы в среде Excel позволяет моделировать различные варианты распределения вкладов и отслеживать их влияние на результат. При этом изменения вносятся в первый и второй столбцы таблицы 3.1. Распределение вкладов автоматически определяется в третьем и четвертом столбцах таблицы, а также на диаграмме Парето.

3.4 Варианты заданий

Построить диаграмму Парето согласно варианту. Данные для вариантов взяты из Интернета и книги[10.2.1].

1. В таблице 3.2 представлены объемы финансовых потерь от различных видов материальных атак для расчета диаграммы Парето.

Таблица 3.2 – Объем потерь от различных видов информационных атак

Порядковый номер	Факторы	Вклад фактора (ден. ед.)
1	Вирусы	15 691 460
2	Неавторизованный доступ к данным	10 617 000
3	Кража ноутбуков и мобильных устройств	6 642 660
4	Кража конфиденциальных данных	6 034 000
5	DOS-атаки	2 922 010
6	Финансовое мошенничество	2 556 900
8	Телекоммуникационное мошенничество	1 849 810
9	Боты внутри организации	1 262 410
10	Проникновение в систему	923 700
12	Проникновение в беспроводную сеть	758 000
13	Неавторизованный доступ к данным	647 510
14	Саботаж	422 500
15	Неправомерное использование web-приложений	269 500
16	Другое	251 310
17	Общая сумма	50 848 770

2. Крупными и средними предприятиями Оренбургской области за 1999 г. произведено продукции на 37997.9 млн. руб. По отраслям промышленности соотношение произведенной продукции характеризуется данными, представленными в таблице 3.3.

Таблица 3.3 – Объем произведенной продукции по отраслям промышленности

Порядковый номер	Наименование отраслей промышленности	Объем произведенной продукции (млн. руб.)
1	Электроэнергетика	3997,7
2	Топливная	17141,9
3	Черная металлургия	6658,7
4	Цветная металлургия	2224,5
5	Химическая и нефтехимическая	893,6
6	Машиностроение	2620,7
7	Деревоперерабатывающая	76,8
8	Промышленность строительных материалов	1152,5
9	Легкая	577,0
10	Пищевая	1884,4
11	Мукомольно-крупяная	698,0

3. В таблице 3.4 представлены мировые объемы заражения пользовательских компьютеров в сети Интернет за октябрь 2010 года. Всего было проверено: 84 690 812 733, из них инфицировано: 32 677 797.

Таблица 3.4 – Вредоносные файлы, обнаруженные в октябре на компьютерах пользователей сети Интернет

Порядковый номер	Наименование вредоносной программы	Количество инфицированных компьютеров
1	Win32.HLLP.Neshta	8277556
2	Win32.HLLP.Whboy.45	6770512
3	Trojan.DownLoader.42350	4116047
4	ACAD.Pasdoc	1616962
5	JS.Nimda	1334519
6	Exploit.Cpllnk	862594
7	Win32.HLLM.Dref	714726
8	Win32.Virut	596437
9	Trojan.Click.64310	469055
10	Win32.HLLP.Rox	458892
11	Win32.Antidot.1	450340

4. В таблице 3.5 представлено количество различных видов дефектов с видеока-

мерами.

Таблица 3.5 – Виды дефектов цифровых видеокамер

Порядковый номер	Наименование дефекта	Число дефектов
1	Плохое закрепление болтов	320
2	Загрязнение направляющих линз	120
3	Внешние царапины	70
4	Неточное соединение деталей	59
5	Неаккуратная сборка узлов	15
6	Дефекты электрических контактов	8
7	Дефекты комплектации	3
8	Загрязнение внешней поверхности	2
9	Прочие дефекты	3

5. Грузооборот по Оренбургской области за 1999 г. представлен в таблице 3.6.

Таблица 3.6 – Объем погрузки по основным видам грузов

Порядковый номер	Наименования видов грузов	Объем погрузки, тыс. тонн
1	Кокс	126,0
2	Нефть и нефтепродукты	4895,0
3	Черные металлы	2097,0
4	Лесные грузы	26,0
5	Химические и минеральные удобрения	29,0
6	Зерно и продукты перемола	919,0
7	Цемент	239,0
8	Лом черных металлов	187,0
9	Комбикорма	25,0
10	Строительные грузы	5553,0

6. Распределение субъектов хозяйственной деятельности по формам собственности в 2000 г. по Оренбургской области представлено в таблице 3.7.

Таблица 3.7 – Распределение субъектов хозяйственной деятельности

Порядковый номер	Субъекты хозяйственной деятельности по формам собственности	Объем субъектов, единиц
1	Государственная	3113
2	Муниципальная	2790
3	Общественных организаций	3490
4	Частная	21983
5	Смешанная российская	1951
6	Иностранных государств	8
7	Иностранных юридических лиц, граждан и лиц без гражданства	89
8	Смешанная с совместным российским и иностранным участием	150

7. В таблице 3.8 представлены факторы, нарушающие работоспособность типографских станков.

Таблица 3.8 – Факторы, нарушающие работоспособность типографских станков

Порядковый номер	Наименование фактора	Количество случаев
1	Дефекты печати	38
2	Некачественная бумага	13
3	Ошибки при верстке	12
4	Повреждение упаковки картриджей	7
5	Неправильный выбор типа бумаги	4
6	Проблемы с типографской краской	3
7	Повреждение при перевозке	3
8	Повреждение на конвейере	2
9	Неправильная балансировка печатных станков	1

8. Распределение населения Оренбургской области по размеру среднедушевого денежного дохода представлено данными таблицы 3.9.

Таблица 3.9 – Распределение населения Оренбургской области по размеру среднедушевого денежного дохода

Порядковый номер	Население со среднедушевыми денежными доходами в месяц, рублей	в % к общей численности населения
1	До 400,0	4,6
2	400,0-600,0	15,5
3	600,0-800,0	20,5
4	800,0-1000,0	18,4
5	1000,0-1200,0	13,8
6	1200,0-1600,0	15,8
7	1600,0-2000,0	6,7
8	Свыше 2000,0	4,7

При необходимости возможно использование произвольных данных по указанию преподавателя.

3.4 Контрольные вопросы

1. В чем заключается суть принципа Парето?
2. Порядок построения диаграммы Парето?
3. Привести собственный пример на применение принципа и диаграммы Парето.
4. В чем заключаются положительные особенности построения диаграммы Парето в среде MS Excel?

5. Какие инструментальные средства могут быть также использованы для анализа Парето кроме табличного процессора MS Excel?

4 Лабораторная работа № 4. Прогнозирование временных рядов на основе уравнений регрессии

Цель работы: Освоить технологию построения регрессионных моделей для прогнозирования временных рядов в среде Excel

4.1 Теоретическая часть

Временной ряд (ВР) $y(t)$ можно интерпретировать в виде суммы двух компонент – детерминированной составляющей $f(t)$ и случайного отклонения $\varepsilon(t)$.

$$\hat{y}(t) = f(t) + \varepsilon(t), \quad (4.1)$$

где $\hat{y}(t)$ - математическая модель временного ряда;

t – порядковый номер элемента ВР, $t=1,2, 3 \dots$

В основе моделирования и прогнозирования ВР лежат операции идентификации (определения) функций $f(t)$ и $\varepsilon(t)$.

Функция $f(t)$ должна иметь такой вид, чтобы сумма квадратов отклонений $\varepsilon(t)$ была минимальной, т.е.

$$\sum_{t=1}^n [y(t) - f(t)]^2 = \sum_{t=1}^n \varepsilon(t)^2 \Rightarrow \min. \quad (4.2)$$

При построении детерминированной и случайной составляющих модели ВР сначала определяют общий вид функций $f(t)$ и $\varepsilon(t)$, а затем – их коэффициенты.

Для определения вида $f(t)$ (иногда ее называют трендом) чаще всего используют следующие функции:

$$f(t) = a_0 + a_1 t, \quad (4.3)$$

$$f(t) = a_0 + a_1 t + a_2 t^2 \quad (4.4)$$

$$f(t) = a_0 + \frac{a_1}{t}, \quad (4.5)$$

где: выражение (4.3) представляет собой полином первой степени (линейная зависимость), (4.4) - полином второй степени (параболическая зависимость), а (4.5) – гиперболическая зависимость.

Вид тренда можно выбрать визуально по графическому отображению $y(t)$. В данной работе принимается гипотеза о параболической зависимости, т.е. $f(t)$ определяется по выражению (4.4). Тогда задача нахождения тренда формулируется следующим образом: найти значения коэффициентов a_0 , a_1 и a_2 в соответствии с выражениями (4.2) и (4.4). Эта задача решается с использованием метода наименьших квадратов (МНК) и инструментальных средств Excel.

После оценки коэффициентов производят экстраполяцию детерминированной основы модели, позволяющую перенести выводы, полученные на участке наблюдения, на явления, находящиеся вне этого участка. Экстраполяция – процедура установления значений ряда в точках, лежащих вне интервала (t_1, t_n) . Экстраполяция дает точечную прогнозную оценку, вычисление которой осуществляется путем решения найденного уравнения регрессии $f(t)$ для значения аргумента t_{n+k} , соответствующего требуемому времени упреждения t_{n+k} . Например, для параболического тренда точечная оценка детерминированной части прогноза \hat{y}_{n+k} вычисляется следующим образом:

$$\hat{y}_{n+k} = a_0 + a_1 t_{n+k} + a_2 t_{n+k}^2. \quad (4.6)$$

Прогнозирование случайной компоненты $\varepsilon(t)$ производится методом авторегрессии. Процессом авторегрессии называется процесс, значения которого в последующие моменты времени зависят от его же значений в предшествующие моменты времени:

$$\varepsilon(t) = b_1 \varepsilon(t-1) + u(t), \quad (4.7)$$

$$\varepsilon(t) = b_1 \varepsilon(t-1) + b_2 \varepsilon(t-2) + \dots + b_n \varepsilon(t-n) + u(t), \quad (4.8)$$

где $b_1 - b_n$ – коэффициенты уравнения авторегрессии;

n – порядок авторегрессии;

$u(t)$ – ошибка авторегрессии.

Расчет коэффициентов $b_1 - b_n$ также производится методом наименьших квад-

ратов. Выбор порядка авторегрессии является одним из этапов построения модели авторегрессии. В настоящей работе задается порядок авторегрессии $n=1$.

Построение прогнозирующей модели временного ряда рекомендуется проводить в три этапа:

- построение детерминированной части модели ВР;
- построение стохастической части модели;
- определение полного прогноза ВР на основе результатов двух предыдущих этапов.

4.2 Порядок выполнения работы

4.2.1 Построение детерминированной части прогнозирующей модели ВР

1. Ввести исходные данные ВР (не менее 20 чисел) в столбец **A** первого листа программы Excel, как показано на рисунке 4.1;

2. Предположим, что исходный временной ряд описывается выражением 4.4. Для построения параболической зависимости необходимо в столбец **B** ввести нумерацию элементов ВР t , а в столбец **C** квадрат t , т.е. t^2 ;

3. Для вычисления коэффициентов модели и дополнительных результатов статистики в правой части экрана с помощью левой кнопки мыши выделить область пустых ячеек размером 5×3 (5 строк и 3 столбца, количество столбцов должно соответствовать количеству оцениваемых коэффициентов).;

3. Активизировать режим вычисления коэффициентов уравнения регрессии в следующем порядке:

- для Excel 2003: «Вставка→Функция→Статистические →Линейн. →Ок»;
- для Excel 2007: перейдите на вкладку «Формулы» и выбрать пункты «Вставить функцию→Функция→Статистические →Линейн. →Ок»;

4. В появившемся окне ввести следующие исходные данные:

- **Известные_значения_y** – диапазон, содержащий данные об объекте (выделить мышью столбец A);
- **Известные_значения_x** – диапазон, содержащий данные времени и квадрата

времени (выделить столбцы **B** и **C**);

- **Константа** – логическое значение, которое указывает на наличие или на отсутствие свободного члена в уравнении 4.4 (если вставить “1”, то свободный член a_0 рассчитывается, если -«0», то свободный член равен 0);

- **Статистика** – логическое значение, которое указывает, выводить дополнительную информацию по регрессионному анализу или нет.

Для подтверждения выбранных параметров нужно нажать комбинацию клавиш <CTRL>+<SHIFT>+<ENTER>.

В рассмотренном примере на рисунке 4.1 данные коэффициенты приняли следующие значения: $a_0= 4.2828$, $a_1= -0.032$, $a_2= 0.0023$. Искомое уравнение регрессии детерминированной части модели (4.4) выглядит следующим образом:

$$\hat{y}_t = 4.283 - 0.032t + 0.0023t^2. \quad (4.9)$$

5. Рассчитать модельные значения y_t в диапазоне $t=1-20$, подставляя в полученное уравнение значения t и t^2 . Все данные в таблице должны быть отцентрированы, дробные числа округлены до третьего знака после запятой. Результаты расчетов примера представлены на рисунке 4.1 в столбце **D** (Y_{np1}).

6. Используя графические инструменты Excel, построить графики исходного ВР Y и прогнозируемого ВР Y_{np1} , рассчитанного по выражению (4.4). Сопоставить сходство графиков. Если они сильно отличаются, то возможна ошибка в расчетах.

7. Рассчитать прогнозные оценки ВР на моменты времени $t=21$; $t=22$; $t=23$. Построить график модельных данных для $t=1,2,3,\dots,23$ (рисунок 4.1 «б»).

4.2.2 Построение стохастической части модели ВР

1. Для каждого наблюдения ряда в столбце **E** рассчитать отклонения $\varepsilon(t)$, как разность между соответствующими данными столбцов **A** и **D** так, как показано на рисунке 4.1 «а»;

2. Для определения коэффициента b_1 уравнения (4.7) расположим в расчетной таблице данные случайной компоненты так, как показано в столбце **F** на рисунке 4.1«а»;

	A	B	C	D	E	F	G	H	I	J	K
1	Y	t	t^2	Y_{np1}	ε_t	ε_{t-1}	ε_{np}	Y_{np2}			
2	4,191	1	1	4,253	-0,062			4,253	0,0023	-0,032	4,2828
3	4,202	2	4	4,228	-0,026	-0,062	-0,039	4,189	0,0004	0,0076	0,0348
4	4,212	3	9	4,207	0,005	-0,026	-0,016	4,191	0,8676	0,0467	#Н/Д
5	4,231	4	16	4,190	0,041	0,005	0,003	4,194	55,684	17	#Н/Д
6	4,168	5	25	4,179	-0,011	0,041	0,025	4,204	0,2433	0,0371	#Н/Д
7	4,216	6	36	4,171	0,045	-0,011	-0,007	4,165			
8	4,201	7	49	4,169	0,032	0,045	0,028	4,196			
9	4,216	8	64	4,170	0,046	0,032	0,020	4,191	0,6257		
10	4,198	9	81	4,176	0,022	0,046	0,029	4,205			
11	4,221	10	100	4,187	0,034	0,022	0,013	4,201			
12	4,218	11	121	4,202	0,016	0,034	0,021	4,224			
13	4,21	12	144	4,222	-0,012	0,016	0,010	4,232			
14	4,222	13	169	4,246	-0,024	-0,012	-0,008	4,239			
15	4,229	14	196	4,275	-0,046	-0,024	-0,015	4,260			
16	4,234	15	225	4,308	-0,074	-0,046	-0,029	4,280			
17	4,265	16	256	4,346	-0,081	-0,074	-0,047	4,300			
18	4,368	17	289	4,388	-0,020	-0,081	-0,051	4,338			
19	4,459	18	324	4,435	0,024	-0,020	-0,013	4,422			
20	4,491	19	361	4,486	0,005	0,024	0,015	4,501			
21	4,631	20	400	4,542	0,089	0,005	0,003	4,545			
22		21	441	4,603		0,089	0,056	4,658			
23		22	484	4,667			0,035	4,702			
24		23	529	4,737			0,022	4,758			
25		24	576	4,810			0,014	4,824			
26											
27											

а)



б)

Рисунок 4.1 – Расчетные данные («а») и графики прогнозирующей модели ВР («б»)

3. Определим коэффициент b_1 модели авторегрессии, для этого повторить пункты 3-4 раздела 4.2.1. с учетом того, что в данном случае определяются коэффициенты уравнения первого порядка. В окно исходных данных вставить следующие значения:

- **Известные значения y** – выделить мышью диапазон ячеек **E3-E21**;
- **Известные значения x** – выделить мышью диапазон ячеек **F3-F21**.

В ячейке **I9** представлено расчетное значение коэффициента $b_1 = 0.6257$. В результате расчетов методом наименьших квадратов уравнение авторегрессии первого порядка имеет вид:

$$\varepsilon(t) = 0.6257 \cdot \varepsilon(t-1). \quad (4.10)$$

Уравнение (4.10) построено без свободного члена b_0 ;

4. В столбце G расчетной таблицы (рисунок 4.1 «а») по выражению (4.10) рассчитать модельные значения случайной компоненты для $t=2,3,4,\dots,21$.

5. Используя выражение (4.10), в ячейках $G23-G25$ рассчитать прогнозные значения случайной компоненты для $t=22,23,24$. При вычислении $\varepsilon(22)$ в ячейке $G23$ использовать значение $\varepsilon(21)$ из ячейки $G22$, при вычислении $\varepsilon(23)$ в ячейке $G24$ использовать значение $\varepsilon(22)$ из ячейки $G24$ и так далее.

4.2.3 Расчет оценок полного прогноза

Расчет оценок полного прогноза производится по выражению (4.1) для $t=21,22,23,24$ в ячейках $H22-H25$ по данным ячеек $D22$ и $G22$, $D23$ и $G23$, $D24$ и $G24$, $D25$ и $G25$. По результатам расчетов, представленных в колонках A, D и H построить графики исходного ВР, прогноза на основе детерминированной модели и графика оценок прогноза с учетом случайной компоненты. На рисунке 4.1 «б» для выбранного примера эти графики обозначены как Y, Y_{np1} и Y_{np2} .

Как видно из рисунка, график Y_{np2} более близок к графику Y , что свидетельствует о повышении точности прогнозных оценок при учете случайной компоненты. Дать анализ графиков, полученных в результате выполнения заданного варианта.

4.3 Контрольные вопросы

1. Привести примеры экономических и технических задач, где нужны прогнозные оценки.
2. Дать характеристику модели прогноза.
3. Как выбираются модели детерминированной и стохастической составляющей прогноза?
4. Описать процесс определения коэффициентов модели в среде Excel.
5. Чем отличаются процедуры интерполяции и экстраполяции ВР?

4.4 Варианты заданий

В таблице 4.1 представлены данные временных рядов для прогнозирования.

Таблица 4.1 – Таблица временных рядов

t\Y	Y ₁ (t)	Y ₂ (t)	Y ₃ (t)	Y ₄ (t)	Y ₅ (t)	Y ₆ (t)	Y ₇ (t)	Y ₈ (t)	Y ₉ (t)	Y ₁₀ (t)
1	4,545	4,100	4,121	4,181	4,152	4,156	4,587	4,301	4,584	4,623
2	4,544	4,215	4,102	4,148	4,159	4,141	4,589	4,303	4,592	4,633
3	4,578	4,228	4,112	4,153	4,164	4,139	4,584	4,316	4,584	4,638
4	4,579	4,213	4,131	4,156	4,165	4,120	4,587	4,304	4,592	4,641
5	4,574	4,235	4,168	4,146	4,166	4,087	4,599	4,316	4,584	4,645
6	4,574	4,233	4,174	4,143	4,169	4,031	4,580	4,200	4,586	4,648
7	4,584	4,251	4,201	4,161	4,167	4,018	4,577	4,206	4,589	4,645
8	4,585	4,225	4,216	4,139	4,151	3,987	4,580	4,200	4,589	4,647
9	4,569	4,245	4,198	4,128	4,153	4,072	4,572	4,313	4,592	4,648
10	4,577	4,253	4,221	4,155	4,132	4,138	4,582	4,300	4,594	4,653
11	4,601	4,259	4,228	4,143	4,135	4,164	4,584	4,309	4,597	4,650
12	4,588	4,243	4,210	4,155	4,131	4,190	4,575	4,289	4,594	4,655
13	4,580	4,261	4,222	4,145	4,099	4,216	4,565	4,316	4,602	4,653
14	4,592	4,245	4,209	4,172	4,103	4,203	4,575	4,323	4,604	4,648
15	4,616	4,276	4,237	4,216	4,096	4,189	4,575	4,343	4,616	4,650
16	4,613	4,280	4,265	4,245	4,083	4,190	4,580	4,358	4,626	4,649
17	4,632	4,274	4,367	4,262	4,057	4,243	4,580	4,353	4,626	4,648
18	4,680	4,292	4,459	4,256	4,062	4,277	4,584	4,361	4,631	4,650
19	4,938	4,289	4,491	4,267	4,009	4,287	4,584	4,376	4,636	4,658
20	4,978	4,113	4,731	4,276	4,013	4,167	4,577	4,311	4,645	4,658

5 Лабораторная работа № 5. Оптимизация топологии распределенной системы сбора и регистрации данных

Цель работы: освоить синергетический алгоритм и программу построения и оптимизации топологии распределенной системы сбора, регистрации и обработки данных от технологических объектов

5.1 Теоретическая часть

Распределенные системы сбора, регистрации и обработки данных (РСОД) являются основными подсистемами систем управления распределенными технологическими объектами, такими, как объекты добычи, переработки и транспортировки нефти и газа. Топология таких систем во многом определяет их технико-экономические

характеристики. Одним из основных критериев оптимизации РСОД является критерий минимальных стоимостных затрат на их разработку и эксплуатацию. В процессе разработки топологии РСОД представляется как система, состоящая из множества источников данных \mathbf{D} , с известными координатами на плоскости, и множества средств обработки данных \mathbf{S} , координаты которых надо определить с учетом частных целевых функций и заданных ограничений, соответствующих каждому источнику данных, и общей целевой функции построения РСОД. Задачи такого типа относятся к категории многовариантных задач, требующих больших временных и стоимостных затрат. Одним из методов их решения является использование синергетического подхода, позволяющего на основе алгоритмов самоорганизации элементов систем определять равновесные состояния систем, удовлетворяющие требованиям частных целевых функций. В работе принята гипотеза соответствия найденного равновесного состояния РСОД минимальным затратам на ее построение.

Затраты на каналы связи определяются различными факторами: выбранной технологией, климатическими условиями в зоне разрабатываемой системы, особенностями рельефа. Одним из основных факторов, влияющим на качество проекта, является расстоянием между узлами РСОД. В связи с этим, стоимостные затраты определяются аналитическими функциями расстояния, а остальные параметры задаются соответствующими коэффициентами. В качестве функций затрат в данной работе предлагается использовать линейные, степенные, экспоненциальные зависимости:

$$c_i(l_i) = k_1 \cdot l_i + b_1, \quad c_i(l_i) = l_i^{k_2} + b_2, \quad c_i(l_i) = e^{k_3 \cdot l_i} + b_3, \quad (5.1)$$

где $k_1, k_2, k_3, b_1, b_2, b_3$ - стоимостные коэффициенты;

l_i – расстояние на местности между источником данных d_i из множества \mathbf{D} и средством обработки s_1 из множества \mathbf{S} .

От выбора координат средства обработки (x_q, y_q) зависят суммарные затраты на каналы связи РСОД.

В основе используемой в работе программы лежит алгоритм определения местоположения средств обработки данных РСОД как точки равновесия векторов сил, имитирующих частные целевые функции затрат на каналы передачи данных от некоторого множества источников данных до средства обработки (СОД). Каждый источ-

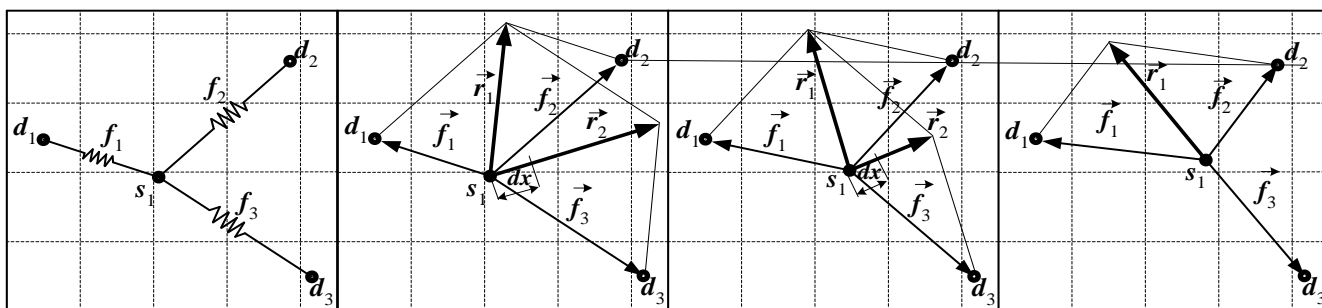
ник данных (ИД) характеризуется частной целевой функцией, определяемой вектором силы, направленным от СОД к источнику данных. Значение модуля силы определяется выбранной функцией затрат от расстояния (5.1). Сила, приложенная к средству обработки со стороны некоторого источника данных, перемещает СОД в направлении к данному ИД, чтобы уменьшить расстояние и, соответственно, сократить затраты на организацию данного канала связи. В процессе итерационного поиска координаты СОД изменяются под воздействием результирующей силы со стороны всех источников данных. Результирующая сила определяется по следующей формуле:

$$\vec{F} = \sum_{i=1}^N \vec{F}_i, |\vec{F}| \rightarrow 0, \quad (5.2)$$

где \vec{F}_i - сила действующая со стороны i -го источника данных.

Модель системы перейдет в состояние равновесия, когда векторная сумма всех сил будет равна нулю. Это состояние соответствует минимальным общим затратам на организацию каналов передачи данных с учетом частных целевых функций.

На рисунке 5.1 представлен порядок определения местоположения СОД для трех источников данных. На рисунке приняты следующие обозначения: d_i - источники данных, f_i - вектора частных целевых функций, r_i – векторная сумма $(i+1)$ -го количества векторов $f_i (i=1, N)$.



а) построение механической модели точки равновесия синергетической системы; б) первый шаг алгоритма поиска координат точки равновесия; в) второй шаг; г) конечный шаг алгоритма

Рисунок 5.1 –Порядок поиска координат точки равновесия модели РСОД

В процессе итерационного поиска точки равновесия координаты СОД изменяются под воздействием результирующей силы со стороны всех ИД. Как видно из рисунка, после перемещения СОД по направлению результирующего вектора r_2 на значение dx , его модуль уменьшился. На последнем шаге итерации представлен результат, где модуль вектора r_2 равен нулю.

5.2 Описание программы

Для выполнения лабораторной работы используется программа «Оптимизация топологии управляющих средств нижнего уровня системы мониторинга распределенных объектов на основе принципа самоорганизации».

На рисунке 5.2 представлено главное окно программы.

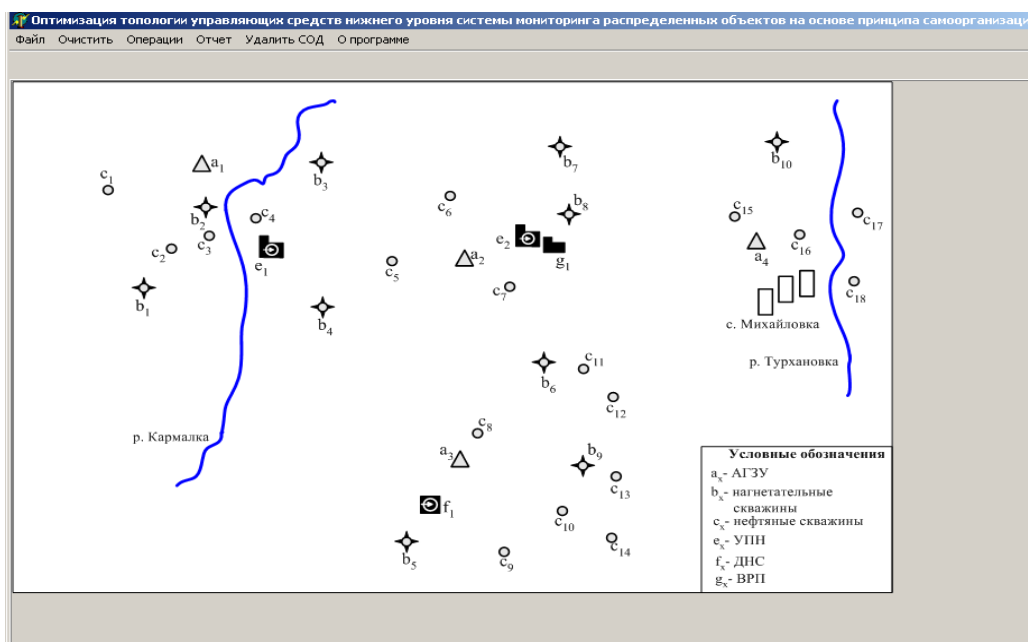


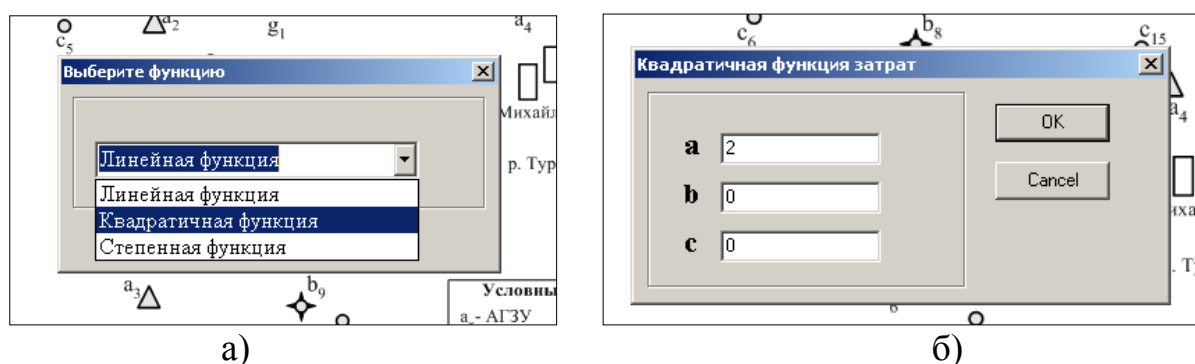
Рисунок 5.2- Главное окно программы

Для начала работы с программой, пользователь должен выбрать карту местности, выполнив команду «Загрузить карту» из меню «Файл», и указать на ней расположение источников данных. Карта должна быть представлена рисунком формата bmp или jpeg. Масштаб карты определяет точность определения координат средства обработки данных с учетом частных целевых функций.

Размещение объектов осуществляется нажатием левой кнопки мыши на плос-

кости карты или загрузкой данных из файла через пункт меню «Файл»-«Открыть». В файле указываются количество источников данных, их координаты на плоскости и тип частной целевой функции. Каждому источнику данных ставится в соответствие аналитическая функция, определяющая зависимость стоимостных затрат на обеспечения требуемого уровня качества канала связи от расстояния.

На рисунке 5.3 представлена экранная форма ввода характеристик частной целевой функции в программный модуль.



а) выбор вида аналитической функции; б) выбор параметров функции
Рисунок 5.3 –Экранные формы ввода параметров аналитических функций

Программа позволяет определять координаты СОД в двух режимах: с использованием алгоритма, основанного на принципах синергетики (самоорганизации), и переборного. В нижней части окна показываються параметры расчета координат СОД: текущие координаты курсора, рассчитанные координаты средства обработки данных X, Y , длина результирующего вектора L , суммарные затраты Z , количество циклов выравнивания Nc , а также задаются шаг итерации и погрешность результирующего вектора.

5.3 Постановка задачи

1. Провести эксперименты с использованием программы для определения координат СОД для различных конфигураций размещения ИД переборным и самоорганизующимся алгоритмом.
2. Сделать анализ результатов экспериментов;
3. Сделать выводы по работе и составить отчет.

Варианты заданий представлены в таблице 5.1.

Таблица 5.1 –Варианты заданий

Номер варианта	k	b	x_2	y_2	A	x_3	y_3	N	dl
1	1	0.5	100	100	300	150	150	9	0.6
2	1,5	1	50	100	300	120	130	9	0.7
3	2	2.5	60	50	320	100	95	10	0.8
4	2	3	40	30	260	120	160	10	0.9
5	2.5	0	50	40	280	200	140	11	1.0
6	2.5	1	120	100	340	130	210	12	1.0
7	3	1.5	200	100	360	250	200	13	1.1
8	3.5	2	120	150	330	140	140	14	1.1
9	0.5	3	30	20	250	180	210	13	1.2
10	1	1.5	50	70	290	250	260	15	1.4
11	1	2	60	50	310	200	150	15	1.3
12	3	2	80	70	320	170	170	16	1.5
13	2.5	1.5	40	60	300	180	170	16	1.6
14	2.5	0.5	30	50	270	190	190	17	1.8
15	0.5	0.5	70	70	350	220	240	17	1.8
16	1.5	1.5	100	50	370	230	200	10	0.4

5.4 Порядок выполнения работы

1. Разместить на плоскости карты 2 источника данных с линейными частными целевыми функциями и координатами (100,100) и (400,100). Коэффициенты целевой функции k и b выбираются из таблицы 5.1 в соответствии с номером варианта. Определить расположение СОД двумя способами: перебором и синергетическим алгоритмом. Шаг итерации задать равным $dx=1$, а точность расчета координат точки равновесия $dl=0.01$. Записать параметры расчета координат средства обработки данных для обоих вариантов: координаты СОД (x_s, y_s), значения длины результирующего вектора L , затрат Z , количества итераций N_C ;

2. Изменить коэффициент k первого источника данных на значение $2 \cdot k$. Произвести заново расчеты и записать результаты. Сравнить результаты для двух вариантов и сделать выводы;

3. Очистить плоскость карты от источников данных. Разместить на плоскости карты 4 точки по вершинам квадрата со стороной A и координатами левой верхней вершины (x_2, y_2). В качестве частной целевой функции выбрать линейную функцию с

коэффициентами 2 и 1. Определить расположение СОД двумя способами, аналогично пункту 1 и записать результаты;

4. Для исходных данных пункта 3 выполнить 10 экспериментов по расчету координат СОД самоорганизующимся алгоритмом с шагами итерации: 1,3,5,7,9,11,13,15,17,19. Построить график зависимости количества итераций поиска точки равновесия от шага. Определить, как влияет шаг итерации dx на сходимость алгоритма и погрешность определения координат точки равновесия dl ;

5. Для исходных данных пункта 3 и шага $dx=1$ выполнить 10 эксперимента с различными значениями заданной погрешности определения координат точки равновесия: 0.1, 0.6, 1.1, 1.6, 2.1, 2.6, 3.1, 3.6, 4.1, 4.6. Определить, как влияет значение заданной погрешности на количество итераций алгоритма;

6. Расположить случайным образом N источников данных и установить значение требуемой точности расчета координат dl . Произвести расчеты и записать результаты.

5.5 Контрольные вопросы

1. Какой обобщенный критерий оценки качества топологии РСОД использован в работе?

2. Опишите алгоритм поиска координат средств обработки данных РСОД, представленный в работе?

3. В чем суть частной целевой функции?

4. Как соотносятся результаты расчетов переборного и синергетического алгоритмов?

5. Как зависит количество итераций алгоритма от шага итерации? Как влияет значение шага итерации на точность и время поиска решения?

6 Лабораторная работа №6. Разработка программы для микроконтроллера ADuC812

Цель работы: приобрести навыки и умения по разработке и отладке программ для управляющих систем на основе микроконтроллера (МК) семейства MCS-51 при помощи стенда SDK1.1

6.1 Теоретическая часть

6.1.1 Характеристика стенда SDK 1.1

Учебный стенд SDK 1.1 предназначен для изучения архитектуры и методов проектирования систем на основе микроконтроллеров. Основой стенда является микроконтроллер ADuC812, ядро которого совместимо с 8051.

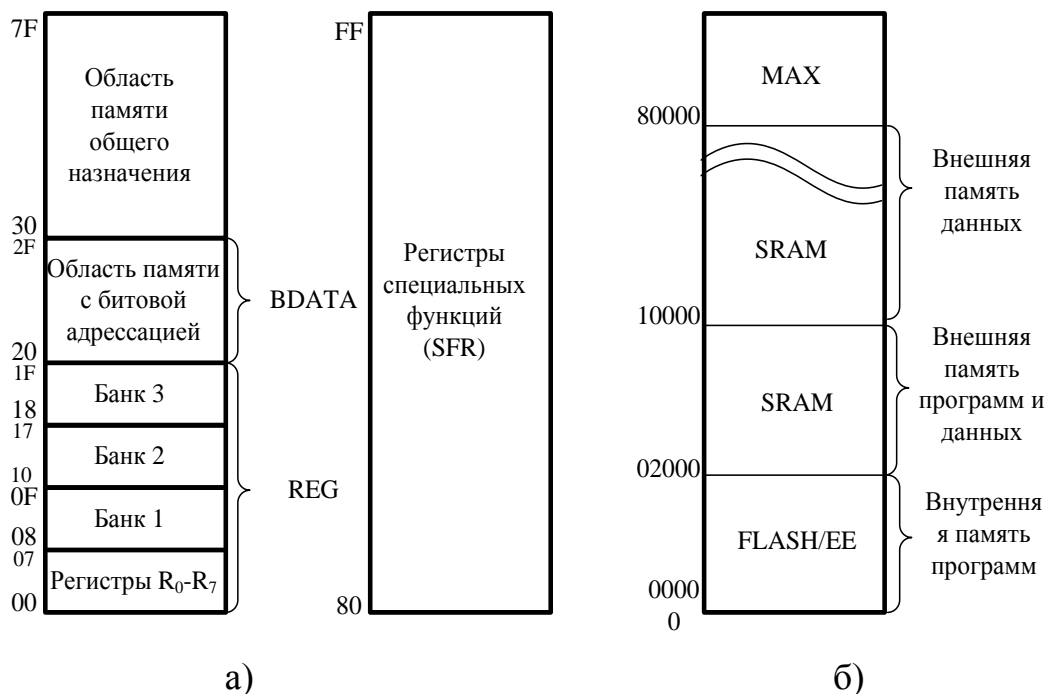
В состав стенда входят: МК ADuC812BS, внешняя E²PROM объемом 256 байт, клавиатура AK1604A-WWB, жидкокристаллический индикатор WH1602B-YGK-CP, часы реального времени PCF8583, 128Кб внешней SRAM, набор сигнальных светодиодов (8 шт.), DIP-переключатели (20 шт.).

В связи с тем, что на стенде установлено большое количество симуляторов сигналов для моделирования различных режимов обмена данными между внешними устройствами (ВУ) и микроконтроллером ADuC812, то их подключение осуществляется не через параллельные порты, а с помощью программируемой логической интегральной схемы МАХ (ПЛИС).

На ПЛИС реализованы регистры, к которым подключены ВУ устройства: светодиоды, звуковой излучатель, клавиатура, 16-разрядный параллельный порт, дисплей. Доступ к данным регистрам осуществляется аналогично, как внешней памяти, т.е. во время цикла обмена по линиям портов *P0* и *P2* передаются адрес регистра и непосредственно данные.

Все МК семейства MCS-51 имеют одинаковую базовую систему памяти: внутреннюю память программ (*Flash*, *EPROM* или *ROM-CODE*), внутреннюю память данных – *DATA*, внешнюю память программ и/или данных (ВПП - *CODE*, ВПД –

XDATA). На рисунке 6.1 приведена карта стандартного распределения памяти стенда. *Flash*-память ADuC812 равна 8КБ (0..1FFF). Внутренняя программная память может отсутствовать. В большинстве случаев ВПП, ВПД ограничены объемом 64КВ (0..FFFF).



а) внутренняя память данных; б) внутренняя память программ и внешняя память программ и данных

Рисунок 6.1 – Распределение памяти стенда SDK 1.1

В тех случаях, когда объема внешней памяти достаточно и для программного кода и для данных, можно использовать совмещенное адресное пространство.

Внутренняя память данных - *DATA* (256 байт) делится на две равные части. В младших 128 байтах (0..7F) располагаются 4 8-ми байтовых банков рабочих регистров – *REG* и область, в которой можно хранить однобитовые переменные – *BDATA*. Далее идет область памяти общего назначения, в которой программист размещает стек, переменные, массивы и другую оперативную информацию. Старшие 128 байтов (80..FF) являются регистрам специальных функций – *SFR* (Special Function Register). *SFR* по сути являются регистрами управления и данных многочисленной периферии МК.

Отладку и пробные пуски программ удобно вести с использованием совмещения ВПД + ВПП. Во флэш-памяти в этом случае располагается резидентный загрузчик пользовательской программы, а во внешнее ОЗУ загружается разрабатываемая

пользовательская программа. Когда пользовательская программа окончательно написана и отлажена ее можно записать во флэш-память МК вместо резидентного загрузчика и/или во внешнее ПЗУ.

6.1.2 Разработка программного кода для стенда в среде Keil uVision

Интегрированная среда Keil uVision поддерживает все стадии разработки приложения: создание исходного файла на С или Ассемблере, трансляцию, исправление ошибок, линкование объектных файлов, тестирование приложения.

Для начала работы нужно создать новый проект. Для этого запустите интегрированную среду «Keil uVision». Обычно при запуске открывается предыдущий проект, поэтому закройте его из основного меню «Project | Close Project».

В основном меню выберите пункт «Project | New Project». Появится диалоговое окно «Create New Project». Перейдите в папку, где планируете сохранить проект.

Введите имя файла проекта, например, Lr1 затем кликните по кнопке «Сохранить». Откроется диалоговое окно «Select Device for Target».

Создадим в проекте основной ассемблерный рабочий файл с расширением *.a. Для этого из пункта основного меню «File | New» создадим текстовый файл с именем «Text1» по умолчанию и сохраним его командой «File | Save As» под именем Lr1.a. Теперь нужно добавить этот файл в проект. Для этого в левом окне на странице «Files» кликаем правой кнопкой сначала по «Source Group 1» и затем в диалоговом окне «Add Files to Group» выбираем файл «Lr1.a».

В результате слева во вкладке «Source Group1» появится файл Lr1.a и на экране будет показано содержимое файла. Далее нужно написать код программы на языке Ассемблера. Пример кода программы, который зажигает светодиоды стенда представлен в листинге программы 1.

Листинг 1. – Пример программы, управляющей светодиодами

```
Name Primer;название программы
DPP EQU 84h ; адрес регистра, который определяет
;номер страницы внешней памяти
SV EQU 7 ; относительный адрес регистра светодиода
WriteSvet macro ; макрос, который зажигает светодиоды в регистре r3 должно
```

```

;содержаться значение, которое будет отображаться на светодиодах
mov r0,DPP ; сохраняем старое значение DPP
mov DPP,#8 ; выбираем 8 страницу внешней памяти
mov r1,#sv ; помещаем в регистр r1 адрес светодиода,
;который располагается на 8 странице
mov a,r3 ; в аккумулятор помещаем значение регистра r3
mov dph,#0 ; старшие разряды адресных линий устанавливаем в ноль
movx @r1,a ; используя косвенную адресацию, помещаем во
;внешнюю память значение аккумулятора
mov DPP,r0 ;восстанавливаем номер страницы внешней памяти
Endm ; макрос закончен
cseg at 0 ; начало сегмента кода, который начинается с адреса 0
jmp start ; переход на начало программы
org 2100h ; дальнейший код будет располагаться по
;адресам, начиная с 2100h
using 0 ; используем нулевой банк ОЗУ
start:
mov r3,#0 ; устанавливаем первоначальное значение регистра r3
light:
Pr_bt: jnb r3.2,pr_bt ; цикл, который ожидает появления на
;2 линии порта r3 единицы
WriteSvet ; зажигаем светодиоды
Inc r3 ; увеличиваем значение регистра r3
mov r4,#3 ; реализуем задержку равную, примерно, 1 с
wait_1: mov r5,#255 ; при помощи пустых циклов
wait_2: mov r6,#255
wait_3: DJNZ r6,wait_3
DJNZ r5,wait_2
DJNZ r4,wait_1
jmp light
end

```

Разработанная программа осуществляет циклический счет в двоичном коде от 0 до 255 и отображает его на светодиодах стенда. Разрешением счета является сигнал «1» на линии *P3.2* параллельного порта МК, который подключен к линии группы переключателей *SW3*.

Директива *CSEG*(Code Segment) задает адрес (начало), с которого будут располагаться коды программы (по умолчанию с нулевого адреса). Команда «*jmp start*» осуществляет переход к адресу, с которого располагается в ОЗУ, разрабатываемая программа, и должна располагаться по адресу 0000h во внутренней флэш-памяти МК. Так как при включении или рестарте МК начинает свою работу именно с этого адреса 0000h.

Следующая директива *ORG* 2100h переопределяет начальный адрес следующего за ней кода программы.

Регистр светодиода находится по адресу 80007h. Для обращения к этому регистру необходимо выбрать 8 страницу, поместив число 8 в регистр *DPP*. Для записи комбинации, высвечиваемой на светодиодах, нужно использовать команду *movx*, которая осуществляет передачу данных во внешнюю память.

Задержка реализуется при помощи пустых циклов. При расчете количества циклов необходимо исходить, что один цикл выполняется примерно за $3 \cdot 10^{-6}$ с при тактовой частоте МК равной 12 МГц.

Для проверки наличия на линии *P3.2* единицы используется команда «*jmp p3.2,pr_bt*», которая осуществляет переход на метку *pr_bt* только в случае, когда контролируемый бит равен нулю.

Завершает программу «*jump light*», который организует бесконечный цикл. Разработчик должен учесть, что МК закончив выполнение программы, будет бесконтрольно выбирать неинициализированные байты из свободной области памяти и выполнять их. Это приведет либо к «зависанию», либо к перезагрузке системы, если МК снова доберется до нулевого адреса. В худшем случае, если последовательность таких байтов образует вредоносный код, то стенд может выйти из строя. Для решения этой проблемы можно, либо оформить программу в виде бесконечного цикла, либо записать в конце программы бесконечный цикл.

После этого скомпилируйте проект, выбрав команду «Project|Build target». Если в программе есть синтаксические ошибки, то о них будут выданы в окне сообщений с краткой их характеристикой и указанием номера строки.

Для загрузки программы в ОЗУ стенда загрузчиком используется файл формата «HEX-80». Необходимо указать данный формат в Keil uVision, выбрав команду «Project| Options for Target» и отметив на вкладке «Output» (рисунок 6.2).

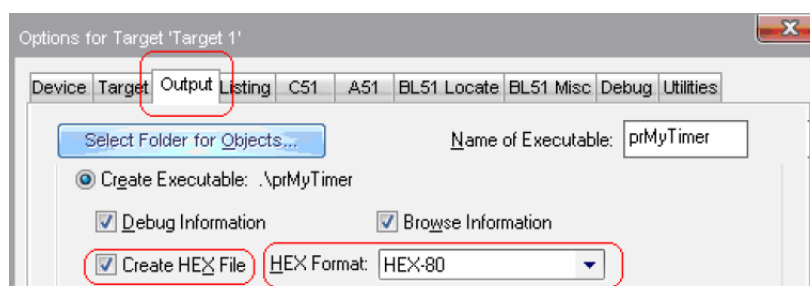


Рисунок 6.2 –Выбор выходного формата файла

Настроить опции проекта, для чего выбрать вкладку «Target» и выставить параметры, как показано на рисунке 6.3.

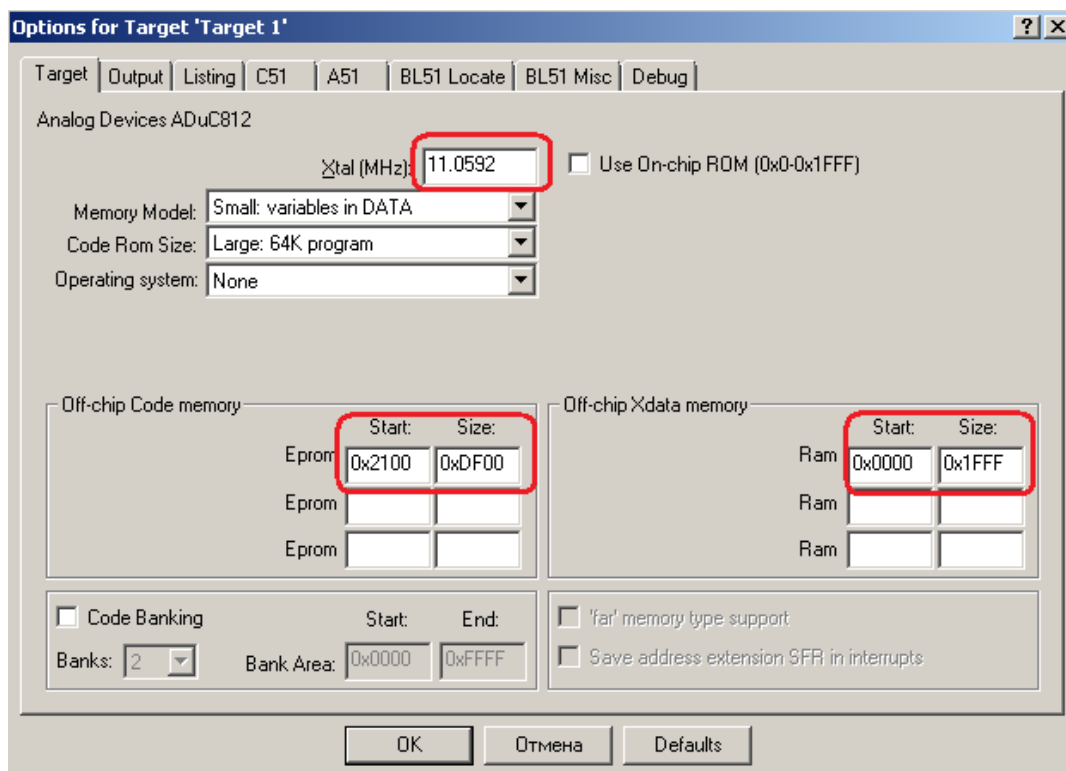


Рисунок 6.3 – Параметры настройки проекта

Теперь, создайте загрузочный HEX-файл программы, для чего нажмите на кнопку в меню «Project | Build Target». Если трансляция и компоновка прошли успешно, то появится сообщение «0 Error(s)».

6.1.3 Загрузка приложения в SDK-1-1 при помощи инструментальной системы T167B

Для подключения стенда SDK 1.1 к компьютеру осуществляется через интерфейс RS232c. Кабель подключается к разъему **JDP1** стенда и COM-порту компьютера. Далее подключите питание к стенду. На ЖК дисплее должна появиться надпись «HEX 202-01. Ожидание загрузки». Если на ЖК дисплее присутствует другая надпись при подаче питания или после нажатия кнопки «RESET», убедитесь, что переключатель 12 группы SW3 находится в состоянии «OFF».

Для загрузки программы в стенд используется программа t2.exe. Эта программа

командной строки и для её запуска удобнее воспользоваться командным файлом. Создайте текстовый файл и сохраните его с расширением bat, например Load.bat.

В файле наберите следующий текст:

```
@echo off  
t2 load.t2
```

Затем создайте текстовый файл load.t2 с текстом:

```
0x2100 0x0 addhexstart Lr1.hex  
opencom1  
loadhex Lr1.hex  
0 term  
bye
```

В первой строке 0x2100 адрес, начиная с которого будет загружено приложение. Во второй строке первая цифра 1 означает, что стенд подключен к COM1. Если стенд подключен к другому порту, то измените первую цифру на его номер. В третьей строке вместо Lr1.hex напишите имя файла, который вы создали. При этом нужно учесть, что в командном файле не используются полные пути к файлу, поэтому нужно убедиться, что все следующие файлы находятся в одной папке: t2.exe, load.bat, load.t2, Lr1.hex.

Дальше нужно перезагрузить стенд, нажатием кнопки RESET, и запустить командный файл load.bat. В результате программа загрузится в SDK 1.1 и микроконтроллер будет осуществлять счет на светодиодах.

6.2 Задание

Разработать программу для микроконтроллера ADuC812 в среде Keil uVision и протестировать на стенде SDK1.1.

Программа должна управлять 8 светодиодами и обрабатывать нажатие первого переключателя SW3. На светодиодах должно отображаться в двоичном коде значения счетчика, который отсчитывает значения с интервалом в 1 секунду.

Переключатель 1 группы SW3 осуществляет разрешение счета в пределах заданного диапазона (нулем или единицей в зависимости от варианта), начиная с минимального значения, и подключен к выводу P3.2. При остановке счета необходимо отобразить на светодиодах значение младшего байта расширенного параллельного порта EXT_LO. Выводы порта EXT_LO подключены к переключателям с 5 по 12

группы SW3

Регистр *r3* используется для хранения значения, которое будет отображаться на светодиодах.

Процедура WriteSvet осуществляет вывод значения регистра *r3* на светодиоды, а ReadPortLo – чтение младшего байта расширенного порта *Ext_LO*.

Варианты заданий представлены в таблице 6.1, а расшифровка кодов – в таблице 6.2.

Таблица 6.1 – Варианты заданий

Вариант	Код направления счета		Код диапазона счета						Код разрешения счета	
	1	2	1	2	3	4	5	6	1	2
1	+		+						+	
2	+		+							+
3	+			+					+	
4	+			+						+
5	+				+				+	
6	+				+					+
7	+					+			+	
8	+					+				+
9	+						+		+	
10	+						+			+
11	+							+	+	
12	+							+		+
13		+	+						+	
14		+	+							+
15		+		+					+	
16		+		+						+
17		+			+				+	
18		+			+					+
19		+				+			+	
20		+				+				+

Таблица 6.2 – Таблица расшифровки кодов задания

Код	Направление счета	Диапазон счета		Разрешение счета
		min	max	
1	Суммирующий	0	32	нулем
2	Вычитающий	4	64	единицей
3		8	128	
4		16	255	
5		4	128	
6		2	96	

6.3 Порядок выполнения работы

1. Создайте отдельную папку для нового проекта Keil uVision и новый проект;
2. Разработайте программу на языке Ассемблера 51. За основу можете взять программу, представленную в листинге 1.

Для проверки достижения максимального (минимального значения) при счете можно использовать следующую команду:

```
CJNE Rn, #direct8, label (6.1)
```

где **Rn** – регистр микроконтроллера;

#direct8 – константа;

Label – метка.

Команда осуществляет переход на метку, если значение регистра не равно константе. Рассмотрим код на языке высокого уровня и его аналог на языке ассемблера 51:

```
If R3<>60 Then          CJNE R3,60, l1
    Inc r3                mov r3,#12
Else r3:=12;            sjmp l2
                        l1: inc r3
                        l2:
```

Для реализации условий больше или меньше можно использовать команду вычитания и команды перехода, анализирующие флаг переноса (**JC**, **JNC**).

Необходимо обратить внимание, что светодиоды и младший байт расширенного параллельного порта подключены не к портам МК, а к регистрам, реализованным на ПЛИС. При обращении к ним нужно использовать команды, использующиеся для внешней памяти и при этом выбирать 8 страницу памяти.

Режим чтения или записи расширенного порта определяется младшим битом регистра **ENA**. Если он равен 1, то **EXT_LO** настроен на чтение данных МК из порта, иначе на запись. Регистры светодиодов, **EXT_LO** и **ENA** имеют адреса 80007h, 80002h и 80004h. Ниже приведен код, который выполняет чтение из **EXT_LO** в регистр микроконтроллера **A**:

```
ReadPortLo:
    mov r0,DPP; сохраняем предыдущее значение DPP
    mov DPP,#8; выбираем 8 страницу внешней памяти
    mov r1,#4; в r1 помещаем относительный адрес ENA
```

```

mov dph,#0
movx a,@r1; в аккумулятор A помещаем содержимое
           ;управляющего регистра ENA
ANL A,#11111110b; обнуляем младший бит регистра A
movx @r1,A; записываем содержимое A в регистр ENA
mov r1,#2 ;в r1 помещаем относительный адрес EXT_LO
movx a,@r1; считываем в A значение порта EXT_LO
mov DPP,r0; восстанавливаем номер страницы
ret      ; выходим из процедуры

```

3. Выполните компиляцию проекта. Исправьте, если есть, синтаксические ошибки;

4. Выполните отладку программы. Для этого выберите команду «Debug|Start/Stop Debug Session». Можно выполнять программу в автоматическом или пошаговом режиме нажав соответственно кнопки **F5** и **F10**.

В режиме отладки слева на экране показано содержимое основных регистров МК, счетчика команд, регистра флагов (рисунок 6.5).

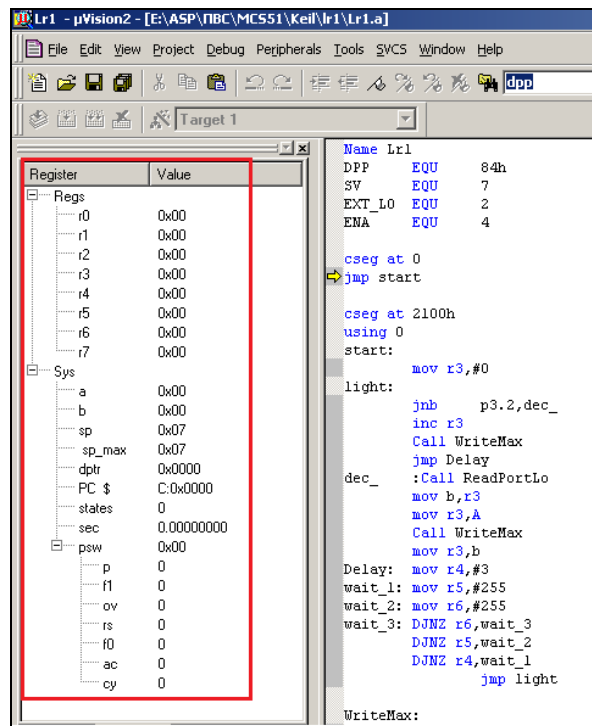


Рисунок 6.5 – Режим отладки программы

При помощи меню «*Peripherals*» можно просматривать и вносить изменение в состояние интегрированной периферии микроконтроллера, например, изменять значение сигнала на выводах порта **P3**.

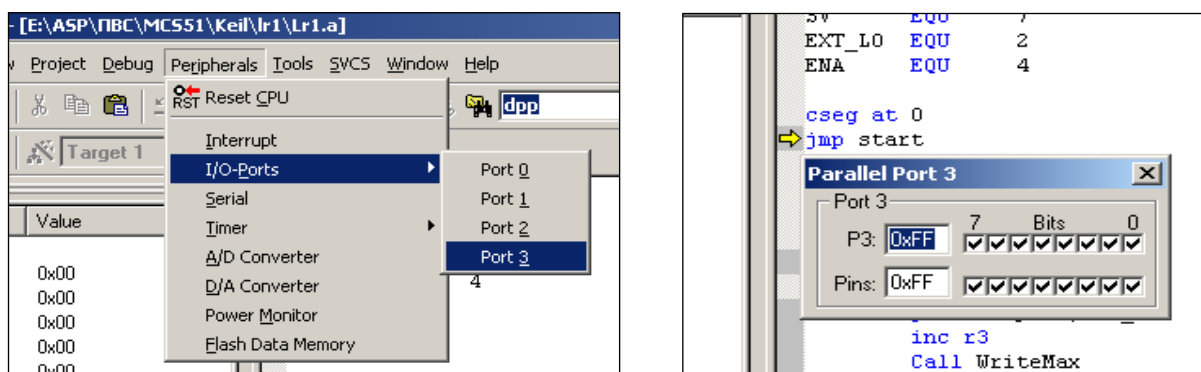


Рисунок 6.6 – Изменение сигнала на выводах порта P3

5. Выполните загрузку программы в стенд;
6. Оформите отчет по работе, сделайте выводы.

6.4 Контрольные вопросы

1. Какими возможностями обладает интегрированная среда Keil uVision?
2. Как организовать задержку во времени при выполнении программы?
3. К какому порту компьютера подключается стенд SDK 1.1?
4. 6. Как распределена память данных в микроконтроллере ADuC812?
5. Как распределена память программ в микроконтроллере ADuC812?

7 Лабораторная работа № 7. Разработка подсистемы ввода-вывода и регистрации сигналов в режиме реального времени

Цель работы: приобрести навыки и умения по использованию таймеров и системы прерываний микроконтроллера ADuC812 для контроля и управления внешними объектами в режиме реального времени

7.1 Теоретическая часть

Для контроля и управления технологическими объектами часто требуется выполнять обработку данных в режиме реального времени. Для этого используются микроконтроллеры, которые характеризуется высокой надежностью и невысокой

стоимостью. Для обеспечения работы в режиме реального времени в микроконтроллеры содержат таймеры и реализуют средства обеспечения системы прерываний. Рассмотрим принципы использования таймеров и прерываний на примере микроконтроллера ADuC812, модификации распространенного МК 8051.

МК ADuC812 имеет три встроенных таймера. Встроенные таймеры чаще всего используются при измерениях различных временных параметров внешних по отношению к микроконтроллеру сигналов, для отработки временных задержек и для выполнения периодических операций.

Таймеры являются 16-разрядными и аппаратно реализованы как два 8-разрядных регистра THx и TLx ($x=0,1,2$ соответственно для таймера 0, 1 и 2), которые могут программироваться отдельно. Все таймеры могут быть настроены на два режима работы: «таймера» и «счетчика».

В режиме «таймера» регистр увеличивается на 1 с частотой $f_k/12$, где f_k – частота МК, а в режиме «счетчика» - по перепаду на соответствующем входе Tx ($x=0,1,2$ соответственно для таймера 0, 1 и 2).

Для конфигурации и контроля таймеров используются 3 регистра специальных функций: $TCON$ и $TMOD$ для управления таймерами 0 и 1, $T2CON$ для управления таймером 2. Более подробную информацию смотрите в следующей литературе [10.2.7].

МК ADuC812 поддерживает 9 источников прерываний с 2 уровнями приоритета. Адреса векторов указаны в таблице 7.1. Конфигурирование и контроль системы прерываний осуществляется через 3 SFR регистра: IE , IP , $IE2$.

Чтобы микроконтроллер начал обрабатывать прерывания, нужно их разрешить при помощи регистра IE . IE (SFR адрес= A8h) - регистр разрешения прерываний, каждый бит которого отвечает за разрешение определенного прерывания.

IP (SFR адрес= B8h) -регистр приоритета прерываний. После включения питания по умолчанию содержит 00h.

Когда происходит прерывание, значение программного счетчика помещается в стек, а в сам счетчик загружается адрес соответствующего вектора прерывания. Прерывания ADuC812 имеют вектора в диапазоне 03h-43h, которые попадают в про-

странство *Flash*-памяти. Данная область не доступна пользователю для записи своих процедур обработки прерываний (точнее, команды перехода к процедурам) по адресам, соответствующим векторам прерываний. В стенде SDK 1.1 (на базе МК ADuC812) реализована следующая схема использования прерываний в пользовательских программах по адресам векторов прерываний во Flash-памяти располагаются команды переходов на вектора пользовательской таблицы, размещенной в диапазоне 2003h-2043h.

Таблица 7.1 – Вектора прерываний

Прерывание	Наименование	Адрес вектора	Приоритет
<i>PSMI</i>	Источник питания ADuC812.	43H	1
<i>IE0</i>	Внешнее прерывание <i>INT0</i> .	03H	2
<i>ADCI</i>	Конец преобразования АЦП.	33H	3
<i>TF0</i>	Переполнение таймера 0.	0BH	4
<i>IE1</i>	Внешнее прерывание <i>INT1</i> .	13H	5
<i>TF1</i>	Переполнение таймера 1.	1BH	6
<i>I²CI/ISPI</i>	Прерывание последовательного интерфейса (<i>I²C, SPI</i>).	3BH	7
<i>RI/TI</i>	Прерывание UART.	23H	8
<i>TF2/EXF2</i>	Переполнение таймера 2.	2BH	9

Например, для прерывания от таймера 0 по адресу 03h находится команда перехода на адрес 2003h, по которому должен находиться пользовательский обработчик. Так как интервал между адресами векторов составляет только 8 байт, то по адресу 2003h находится команда перехода в другую область.

7.2 Задание

Разработать программу для микроконтроллера ADuC812 в среде Keil uVision и протестировать на стенде SDK1.1. Программа реализует два режима работы. В первом режиме на светодиодах должно отображаться в двоичном коде значения счетчика, который отсчитывает значения с интервалом в 1 секунду.

Во втором, отобразить на светодиодах в двоичном коде количество задних фронтов (перепадов «1»-«0») на линии *Tx* ($x=0$ или 1 в зависимости от варианта), которые соответствуют 3 и 4-му переключателю группы *SW3*, на текущий момент с интервалом в 1 секунду. Первый режим выбирается обработкой внешнего прерывания от переключателя 1 (*INT0*), а второй - 2 (*INT1*) по заднему фронту сигнала.

Задержку реализовать при помощи таймера и обработки прерывания его переполнения. Варианты заданий представлены в таблице 7.2, а расшифровка кодов – в таблице 7.3.

Таблица 7.2 – Варианты заданий

Вариант	Код направления счета		Код диапазона счета						Линия Тх	
	1	2	1	2	3	4	5	6	0	1
1	+		+						+	
2	+		+							+
3	+			+					+	
4	+			+						+
5	+				+				+	
6	+				+					+
7	+					+			+	
8	+						+			+
9	+							+	+	
10	+							+		+
11	+								+	+
12	+								+	+
13		+	+						+	
14		+	+							+
15		+		+					+	
16		+		+						+
17		+			+				+	
18		+			+					+
19		+				+			+	
20		+					+			+

Таблица 7.3 - Таблица расшифровки кодов задания

Код	Направление счета	Диапазон счета	
		min	max
1	Суммирующий	0	32
2	Вычитающий	4	64
3		8	128
4		16	255
5		4	128
6		2	96

7.3 Порядок выполнения работы

1. Создайте отдельную папку и создайте новый проект в среде Keil uVision ;
2. Разработайте программу на языке Ассемблера 51. За основу можете взять код программы, представленный в приложении А.

Задержка в 1с реализована с помощью таймера 2, который переполняется за 50 000 тактов. Обработчик прерывания таймера 2 увеличивает значения регистра *r7*. При частоте МК 12 МГц, чтобы обеспечить задержку в 1 с необходимо подождать, пока значение *r7* не станет равным 20.

3. Программа выполняет два основных действия:

- осуществляет в двоичном коде счет на светодиодах с интервалом в 1с;
- осуществляет подсчет переключению на внешнем входе *Tx* и выводит на светодиоды в двоичном коде.

Для определения режима работы программы можно использовать один из не использованных свободных регистров, например, *r6*. Если его значение равно «0», то команда осуществляет первое действие, иначе – второе. Первоначально проинициализируем значение регистра нулем:

```
-----  
START:  
    mov r3,#0; инициализация значения счетчика  
    mov r7,#0  
    mov r6,#0  
-----
```

4. Изменение значение регистра *r6* будет осуществляться по внешним прерываниям *INT0* и *INT1*. Соответственно, в программе должны быть разрешены данные прерывания:

```
-----  
START:  
    mov r3,#0; инициализация значения счетчика  
    mov r7,#0  
    mov r6,#0  
    setb EX0  
    setb EX1  
-----
```

5. Внешние прерывания *INT0* и *INT1* могут быть вызваны уровнем или переходом сигнала из 1 в 0 на входах микроконтроллера в зависимости от значений управ-

ляющих битов *IT0* и *IT1* в регистре *TCON*. Если соответствующий бит *ITx* ($x = 0,1$) установлен, то прерывание вызывается по перепаду 1–0 на соответствующем входе *INTx* микроконтроллера. Если эти биты сброшены, то прерывание *INTx* вызывается при наличии низкого уровня на соответствующем входе:

```

-----
START:
    mov r3,#0; инициализация значения счетчика
    mov r7,#0
    mov r6,#0
    setb EX0
    setb EX1
    SETB IT0
    SETB IT1
-----

```

6. Укажем в начале программы где находятся подпрограммы обработки внешних прерываний *INT0* и *INT1*:

```

-----
org 02Bh
jmp Timer0
org 2003h
jmp Int0_ ; переход на подпрограмму обработки прерывания INT0
org 2013h
jmp Int1_ ; переход на подпрограмму обработки прерывания INT1
-----

```

7. Реализуем в конце программы подпрограмму обработки прерывания *int0*:

```

-----
Timer0:
    inc r7
    CLR TF2
    reti
INT0_:
    Mov r6,#0; переходим в первый режим работы программы
    Clr TRx; отключаем работу таймера x, который осуществляет количество
;внешних импульсов Tx
    Reti ; возврат к выполнению основной программы
-----

```

8. Реализуем в конце программы подпрограмму обработки прерывания *int0*:

```

-----
INT0_:
    Mov r6,#0; переходим в первый режим работы программы
    Reti ; возврат к выполнению основной программы
INT1_:
    Mov r6,#1; переходим во второй режим работы программы
    mov THx,#0;обнуляем таймер/счетчик, где X- номер таймера
    mov TLx,#0;
    setb TRx; включаем таймер x
    reti; возврат к выполнению основной программы
-----

```

9. Инициализируем таймер x на режим работы как 16-разрядный счетчик для подсчета внешних импульсов по входу Tx :

```

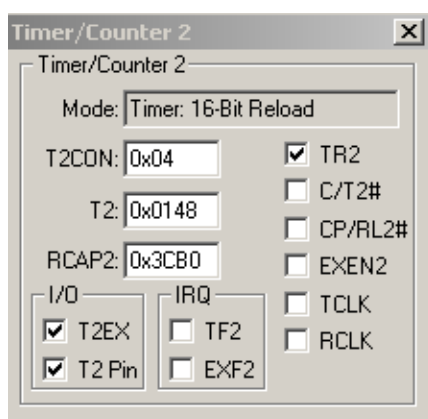
-----
SETB IT0
SETB IT1
mov TMOD,#a; где a,b –значение, которые инициализируют
mov TCON,#b; регистры TMOD и TCON
-----

```

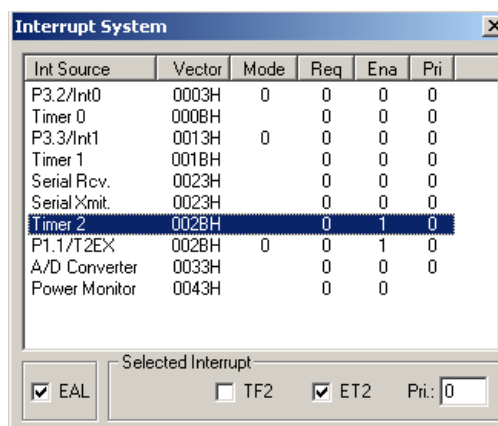
10. Выполните компиляцию проекта. Исправьте, если есть, синтаксические ошибки;

11. Выполните отладку программы. Для этого выберите команду «Debug|Start/Stop Debug Session». Можно выполнять программу в автоматическом или пошаговом режиме нажав соответственно кнопки F5 и F10;

12. Для того, чтобы видеть параметры таймера 2 при отладке, выберите команду «Peripherals|Timer|Timer2».



а)



б)

а) модель таймера 2 в режиме отладке; б) параметры прерывания по таймеру 2

Рисунок 7.2 – Модели аппаратных средств МК

В результате появится окно с параметрами таймера 2 (рисунок 7.2 а);

13. Для просмотра параметров прерывания по переполнению таймера 2 выполните команду «Peripherals|Interrupt» и в появившемся окне выделите строку «Timer 2» (рисунок 7.2 б).

14. Выполните загрузку программы в стенд;

15. Оформите отчет по работе, сделайте выводы.

7.4 Контрольные вопросы

1. Какие регистры специальных функций используются для конфигурирования таймеров/счетчиков?
2. С какой частотой увеличивает значение таймер/счетчик в режиме «таймера»?
3. Какое событие таймера/счетчика может вызвать прерывание работы МК?
4. Как устанавливаются приоритеты прерываний?
5. Как реализована система обработки прерываний на стенде SDK 1.1?

8 Лабораторная работа №8. Методы защиты информации.

Шифр Цезаря

Цель работы: освоить технологию шифрования и дешифрования информации в среде Excel с использованием шифра Цезаря

8.1 Теоретическая часть

Шифр Цезаря является частным случаем шифра простой замены (одноалфавитной подстановки). Свое название этот шифр получил по имени римского императора Гая Юлия Цезаря, который использовал этот шифр при переписке. При шифровании исходного текста каждая буква заменяется другой буквой того же алфавита по следующему правилу. Заменяющая буква определяется путем смещения по алфавиту к концу от исходной буквы на k букв. При достижении конца алфавита выполняется циклический переход к его началу.

Например: пусть A – используемый алфавит:

$$A = \{a_1, a_2, \dots, a_m, \dots, a_N\}, \quad (8.1)$$

где $a_1, a_2, \dots, a_m, \dots, a_N$ – символы алфавита;

N - ширина алфавита.

Пусть k – число позиций сдвига символов алфавита при шифровании, $0 < k < N$.

При шифровании каждый символ алфавита с номером m из кодируемого текста заменяется на символ этого же алфавита с номером $m+k$. Если $m+k > N$, номер символа в алфавите A определяется как $m+k-N$.

Для дешифрования текстовой информации номер позиции символа восстанавливаемого текста определяется как $m-k$. Если $m-k < 0$, то вычисление этого номера производится как $m-k+N$.

Достоинством этой системы является простота шифрования и дешифрования. К недостаткам системы Цезаря следует отнести:

- подстановки, выполняемые в соответствии с системой Цезаря, не маскируют частот появления различных букв исходного и открытого текста;
- сохраняется алфавитный порядок в последовательности заменяющих букв; при изменении значения k изменяются только начальные позиции такой последовательности;
- число возможных ключей k мало;
- шифр Цезаря легко вскрывается на основе анализа частот появления букв в шифре.

8.2 Порядок выполнения работы

1. Создать новый документ в среде Excel, перейти на второй лист этого документа. Начиная с ячейки $A1$ до $A40$ набрать алфавит, как показано на рисунке 8.1. Выделить все символы алфавита и, нажав правую кнопку мыши, назначить имя диапазона «**АВС**» командой «Имя диапазона»;

2. На первом листе документа в ячейке $B1$ набрать текст, который необходимо зашифровать, например: **Гай Юлий Цезарь: «Пришел, увидел, победил!»**.

При наборе текста необходимо использовать только те символы, которые входят в алфавит;

3. В ячейке $B3$ записать формулу «=ПРОПИСН($B1$)», которая переводит символы в строке в прописные буквы;

4. В ячейке $D3$ записать формулу «=ДЛСТР($B3$)», которая позволяет определить длину строки, что необходимо пользователю, для кодировки исходной строки;

5. В ячейку **D4** записать значение ключа шифрования k , например, 5;

6. В столбце **A**, начиная с ячейки **A6**, пронумеровать ячейки числами последовательного ряда от 1 до N , где N – число символов в тексте, включая пробелы. Значение N рассчитано в ячейке **D3**;

7. В ячейку **B6**, записать формулу «=ПСТР(B\$3;A6;1)», которая разделяет кодируемый текст на отдельные символы. Скопировать эту формулу в ячейки **B7-B47**;

8. В ячейку **C6** записать формулу «=ПОИСКПОЗ(B6;ABC;0)». Функция ПОИСКПОЗ производит поиск индекса (номера позиции) символа в массиве **ABC**, который был определен на листе 2. Скопировать содержимое ячейки **C6** в ячейки **C7-C47**;

	A	B	C	D	E
1	.				
2	,				
3	:				
4	:				
5	"				
6	!				
7	;				
8	A				
9	B				
10	B				
11	Г				
12	Д				
13	Е				
14	.				
15	Ж				
16	З				
17	И				
18	Й				
19	К				
20	Л				
21	М				

а)

	A	B	C	D	E	F	G	H
1		Гай Юлий Цезарь:"Пришел, увидел, победил!"						
2								
3		ГАЙ Ю.	42					
4				5				
5								
6	1	Г	11	16	З	З		
7	2	А	8	13	Е	ЗЕ		
8	3	Й	18	23	О	ЗЕО		
9	4		3	8	А	ЗЕОА		
10	5	Ю	39	4	:	ЗЕОА:		
11	6	Л	20	25	Р	ЗЕОА:Р		
12	7	И	17	22	Н	ЗЕОА:РН		
13	8	Й	18	23	О	ЗЕОА:РНО		
14	9		3	8	А	ЗЕОА:РНОА		
15	10	Ц	31	36	Ы	ЗЕОА:РНОАЫ		
16	11	Й	13	18	Й	ЗЕОА:РНОАЙЙ		

б)

	A	B	C	D	E	F	G	H
1		ЗЕОА:РНОАЫЙМЕХЯБВФХН						
2								
3		ЗЕОА:!	42					
4				5				
5								
6	1	З	16	11	Г	Г		
7	2	Е	13	8	А	ГА		
8	3	О	23	18	Й	ГАЙ		
9	4	А	8	3		ГАЙ		
10	5	:	4	39	Ю	ГАЙ Ю		
11	6	Р	25	20	Л	ГАЙ ЮЛ		
12	7	Н	22	17	И	ГАЙ ЮЛИ		
13	8	О	23	18	Й	ГАЙ ЮЛИЙ		
14	9	А	8	3		ГАЙ ЮЛИЙ		
15	10	Ы	36	31	Ц	ГАЙ ЮЛИЙ Ц		
16	11	Й	13	18	Е	ГАЙ ЮЛИЙ ЦЕ		

в)

	D	E	F	G	H	I
33	17	И	ГАЙ ЮЛИЙ ЦЕЗАРЬ:"ПРИШЕЛ, УВИ			
34	12	Д	ГАЙ ЮЛИЙ ЦЕЗАРЬ:"ПРИШЕЛ, УВИД			
35	13	Е	ГАЙ ЮЛИЙ ЦЕЗАРЬ:"ПРИШЕЛ, УВИДЕ			
36	20	Л	ГАЙ ЮЛИЙ ЦЕЗАРЬ:"ПРИШЕЛ, УВИДЕЛ			
37	2	,	ГАЙ ЮЛИЙ ЦЕЗАРЬ:"ПРИШЕЛ, УВИДЕЛ,			
38	3		ГАЙ ЮЛИЙ ЦЕЗАРЬ:"ПРИШЕЛ, УВИДЕЛ,			
39	24	П	ГАЙ ЮЛИЙ ЦЕЗАРЬ:"ПРИШЕЛ, УВИДЕЛ, П			
40	23	О	ГАЙ ЮЛИЙ ЦЕЗАРЬ:"ПРИШЕЛ, УВИДЕЛ, ПО			
41	9	Б	ГАЙ ЮЛИЙ ЦЕЗАРЬ:"ПРИШЕЛ, УВИДЕЛ, ПО			
42	13	Е	ГАЙ ЮЛИЙ ЦЕЗАРЬ:"ПРИШЕЛ, УВИДЕЛ, ПО			
43	12	Д	ГАЙ ЮЛИЙ ЦЕЗАРЬ:"ПРИШЕЛ, УВИДЕЛ, ПО			
44	17	И	ГАЙ ЮЛИЙ ЦЕЗАРЬ:"ПРИШЕЛ, УВИДЕЛ, ПО			
45	20	Л	ГАЙ ЮЛИЙ ЦЕЗАРЬ:"ПРИШЕЛ, УВИДЕЛ, ПО			
46	6	!	ГАЙ ЮЛИЙ ЦЕЗАРЬ:"ПРИШЕЛ, УВИДЕЛ, ПО			
47	5	"	ГАЙ ЮЛИЙ ЦЕЗАРЬ:"ПРИШЕЛ, УВИДЕЛ, ПО			
48						

г)

а) алфавит символов шифра Цезаря; б) начальная часть документа шифрования;

в) и г) начальная и конечная часть документа дешифрования

Рисунок 8.1 – Фрагменты документов Excel по лабораторной работе № 8

9. Получив номер символа в алфавите **ABC**, произвести сдвиг нумерации алфавита для кодируемой последовательности символов. В ячейку **D6** записать формулу:

$$\text{«=ЕСЛИ(ПОИСКПОЗ(В6;ABC;0)+D4>40;ПОИСКПОЗ(В6;ABC;0)+D4-40;ПОИСКПОЗ(В6;ABC;0)+D4)»}. \quad (8.2)$$

Эта формула производит сдвиг номеров символов алфавита на величину k и определяет номер заменяющего символа из алфавита **ABC**. Содержимое **D6** скопировать в область **D7-D47**;

10. Выбрать символы из алфавита **ABC** в соответствии с новыми номерами. В ячейку **E6** записать формулу «=ИНДЕКС(ABC;D6)». Скопировать содержимое ячейки **E6** в область **E7-E47**;

11. Для получения строки закодированного текста необходимо в ячейку **F6** записать «=E6», в ячейку **F7** соответственно – «=F6&E7». Далее скопировать содержимое ячейки **F7**, в область **F8-F47**. В ячейке **F47** прочитать зашифрованный текст;

12. На третьем листе выполнить дешифрование аналогично пунктам 2-11 лабораторной работы. При этом необходимо учесть следующие особенности:

- в п. 2 набрать зашифрованный текст;
- в п. 9 в ячейку **D6** записать формулу:

$$\text{=ЕСЛИ(ПОИСКПОЗ(В6;ABC;0)-D4<0;ПОИСКПОЗ(В6;ABC;0)-D4+40;ПОИСКПОЗ(В6;ABC;0)-D4)}. \quad (8.3)$$

Получение исходного текста в ячейке **F47** третьей страницы свидетельствует о корректном выполнении лабораторной работы.

8.3 Контрольные вопросы

1. В чем заключается принцип защиты информации с использованием шифра Цезаря?
2. Объяснить формулы (8.2) и (8.3).
3. Можно ли использовать программы настоящей лабораторной работы для шифрования другой информации? Если да, то, каким образом?

4. В чем достоинства и недостатки шифра Цезаря?
5. Какие еще методы защиты информации Вам известны?

9 Лабораторная работа №9. Автоматизированное проектирование вычислительной системы с использованием инструментальных средств класса SCADA - «Trace Mode 6.0»

Цель работы: получить общее представление о возможностях SCADA-систем для построения систем контроля и управления с использованием интегрированной среды «Trace Mode 6.0»

9.1 Теоретическая часть

Для автоматизации управления технологическими процессами на базе средств вычислительной техники используются SCADA-системы. Понятие «SCADA» обычно относится к централизованным системам контроля и управления технологическими объектами, расположенными на большой территории. Современные SCADA-системы состоят из трех основных структурных элементов:

- Remote Terminal Unit (RTU), удаленный терминал, осуществляющий перво-степенные задачи управления в режиме реального времени;
- Master Terminal Unit (MTU), диспетчерский пункт управления, осуществляющий контроль и управление высокого уровня, на основе данных, полученных от RTU;
- Communication System (CS), коммуникационная система (каналы связи), необходимая для передачи данных с удаленных точек (объектов, терминалов) на центральный пульт оператора-диспетчера и передачи сигналов управления на RTU

SCADA-системы осуществляют сбор информации о технологических параметрах, сохранение информации в архивах, графическое представление технологического процесса, оповещение персонала об аварийных ситуациях на производстве, автоматическое управление ходом технологического процесса и ряд других.

Процедуры сбор данных начинаются в RTU и включает в себя: регистрацию

показаний измерительных приборов и составление отчетов об отказе оборудования. Далее данные собираются и форматируются таким способом, чтобы оператор диспетчерской, используя человеко-машинный интерфейс, мог принять управляющие решения.

Для знакомства с системой SCADA выбрана интегрированная среда разработки «Trace Mode 6.0», возможности которой характерны для большинства современных подобных систем и которая имеет мощные средства для создания распределенных иерархических АСУТП.

Trace Mode (*произносится «Трэйс мóуд» или «Трэйс мóд»*) — инструментальный программный комплекс класса SCADA/ММІ, разработан компанией AdAstra Research Group в Москве в 1992 году. Комплекс предназначен для разработки программного обеспечения АСУТП, систем телемеханики, автоматизации зданий, систем учёта электроэнергии (АСКУЭ, АИИС КУЭ), воды, газа, тепла, а также для обеспечения их функционирования в реальном времени. Начиная с версии 4.20 (1995 г.) Trace Mode обладает функциями для программирования промышленных контроллеров [10.2.3].

Trace Mode 6.0 имеет встроенные бесплатные драйверы к более чем 1600 контроллерам и платам ввода/вывода, свыше 600 анимационных объектов, более 150 алгоритмов обработки данных и управления, комплексные технологические объекты [10.2.2]. Все это позволяет разрабатывать АСУТП для различных областей промышленности.

Основными понятиями для среды разработки Trace Mode являются:

- **проект** — математические и графические элементы системы, которые функционируют на различных операторских станциях и контроллерах, входящих в одну АСУ ТП и объединенных информационными связями и системой архивирования;
- **узел** — любое устройство в рассматриваемом проекте на котором запущено программное обеспечение Trace Mode. Узлом может быть как станция оператора, так и микроконтроллер, осуществляющий сбор информации или управляющий технологическим процессом;
- **канал** — информационная структура, которая включает в себя переменные,

константы, методы формирования и преобразования значений переменных.

Обмен данными между всеми узлами системы производится при помощи каналов. Каналы делятся на входные (IN) и выходные (OUT). Входной канал получает значение от внешних источников (от микроконтроллера, платы ввода/вывода, к примеру) или от системной переменной (длина архива к примеру).

Данные из внешних устройств записываются в каналы. Данные из каналов посылаются на внешние устройства. В каналы оператор заносит управляющие сигналы. Значения из каналов записываются в архивы, операторские отчеты и т.п. В каналах осуществляется преобразование данных. Меняя значения на системных каналах, можно управлять выводимой на экран информацией, звуковыми сигналами и т.д., т.е. всей системой [10.2.2].

9.2 Постановка задачи

1. Необходимо разработать проект «АРМ диспетчера», контролирующего удаленно один параметр от RTU. АРМ диспетчера и удаленный терминал расположены в разных зданиях. В качестве канала связи используется локальная сеть;
2. Сделать по работе выводы и составить отчет;
3. Ответить на контрольные вопросы.

В таблице 9.1 представлены варианты заданий на лабораторную работу.

Таблица 9.1 – Варианты заданий

Задание Вариант	Код параметра				Код вида прибора индикации		Код типа сигнала				Диапазон сигнала	
	1	2	3	4	1	2	1	2	3	4	min	max
1	+				+		+				-20	20
2	+				+			+			-25	25
3	+				+				+		-30	35
4	+					+				+	-40	40
5	+					+	+				-45	45
6	+					+		+			-50	55
7		+			+				+		-60	60
8		+			+					+	-65	65
9		+			+		+				-20	20
10		+				+		+			-25	25

11		+				+			+		-30	35
12		+				+				+	-40	40
13			+		+		+				-45	45
14			+		+			+			-50	55


Продолжение таблицы 9.1

Задание Вариант	Код параметра				Код вида прибора индикации		Код типа сигнала				Диапазон сигнала	
	1	2	3	4		1	2	3	4		1	2
15			+		+				+		-60	60
16			+			+				+	-65	65
17			+			+	+				-20	20
18			+			+		+			-25	25
19				+	+				+		-30	35
20				+	+					+	-40	40

Таблица 9.2 - Таблица расшифровки кодов задания


Код	Параметр	Вид прибора индикации	Тип сигнала
1	Давление	Ползунок	Синусоидальный
2	Температура	Стрелочный прибор	Треугольник
3	Скорость		Пила
4	Частота		Случайный

9.3 Порядок выполнения работы

1. Создайте новый проект командой «Новый» из меню «Файл». В открывшемся диалоговом окне выберите стиль разработки «Комплексный»;
2. Выберите раздел «Топология». Создайте, нажатием ПК мыши, группу «Объект», а в ней две группы – «Здание_1» и «Здание_2»;
3. Создайте в объекте «Здание_1» канал класса «FLOAT». Перейдите в режим редактирования и установите галочку «Использовать» в разделе «Граница». Выставьте произвольные пороговые значения с учетом диапазона сигнала из таблицы 9.1. Нажмите кнопку «Архивация» и выставьте галочку «Отчет Тревог»;
4. Создайте графический экран узла АРМ. Для этого в разделе «Шаблоны_экран» создайте компонент «Экран». Разместите на верхнем левом углу статический текст, соответствующий параметру из таблицы 9.1 согласно варианту, нажав на кнопку графического элемента «Текст» ;
5. Создайте динамический текст для индикации значения канала. Для этого разместите графический элемент (ГЭ) «Текст» справа от предыдущей надписи и от-


кройте свойства этого ГЭ. Двойным щелчком ЛК на строке «Текст» откройте меню «Вид индикации». В правом поле строки щелчком ЛК вызовите список доступных типов динамики атрибута и из всех предлагаемых типов выберите «Значение»;

6. Выполните привязку ГЭ «Текст» к аргументу экрана. Для этого выполните щелчок ЛК в правом поле строки «Привязка». В открывшемся окне нажатием кнопки INS создайте аргумент шаблона экрана. Измените имя аргумента (должен соответствовать параметру из таблицы 9.1 согласно варианту);

7. Разместите на экране для отображения аргумента ползунков или стрелочный прибор (в зависимости от варианта). Для этого на панели графического редактора нажмите иконку  и выберите из появившегося меню иконку стрелочного прибора или ползунка. Перейдя в режим редактирования свойств прибора, щелчком ЛК на экранной кнопке «Основная привязка» откройте окно табличного редактора аргументов шаблона экрана и выберите ранее созданный аргумент. Для элемента «стрелочный прибор» измените заголовок на значение, которое было установлено у первого ГЭ «Текст». Зайдите в свойство «Полоса» двойным щелчком ЛК и измените верхний и нижний пределы шкалы в соответствии со значениями диапазона сигнала из таблицы 9.1;

8. Откройте вкладку «Источники/Приемники» и ПК создайте в нем группу компонентов «Генераторы». В группе «Генераторы_1» через ПК создайте компоненту сигнала согласно варианту из таблицы 9.1;

9. Создайте в объекте «Здание_2» канал класса «FLOAT». Перейдите в режим редактирования и установите галочки «Использовать» и «Масштабирование» в разделе обработка. В поля «Max» и «Min» входных данных IN введите значения 100 и 0 соответственно. В поля «Max» и «Min» выходных данных A установите значения диапазона сигнала из таблицы 9.1. Нажмите кнопку рассчитать;

10. Откройте дополнительное окно навигатора, нажав ЛК на иконке . В первом окне выделите «Здание_2», а во втором – группу «Генераторы». Перетащите соответствующий источник на «канал_2». Аналогично создайте связь «Канал_2» - «Канал_1»;

11. В разделе «Система» создайте узел «RTM», который представляет собой

компьютер диспетчера. В режиме редактирования узла «RTM_1» введите IP-адрес компьютера диспетчера, установите галочки «Прием» и «Посылка» в разделе «Адаптеры». На вкладке «Отчет тревог/Дамп/Параметры» введите имя файла для сохранения отчета тревог, установите количество записей и состояние в значение «True»;

12. В разделе «Система» создайте узел «MicroRTM», который представляет собой компьютер, генерирующий данные. В режиме редактирования узла «MicroRTM_1» введите IP-адрес компьютера, установите галочки «Прием» и «Посылка» в разделе «Адаптеры»;

13. Распределите каналы по узлам. Для этого перетащите группы здания из раздела «Топология» в соответствующие узлы: «Здание_1» в «RTM_1», а «Здание_2» в «MicroRTM_2»;

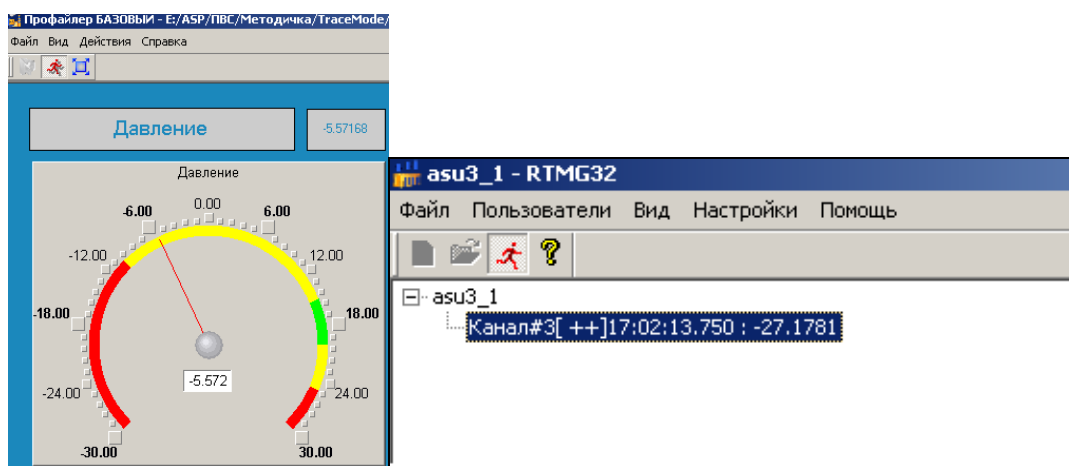
14. Перетащите экран из «Шаблоны экрана» в группу каналов «Здание_1» узла «RTM_1». Выделите объект «Экран_1» и откройте его свойства. Перейдите на вкладку «Аргументы» и выберите параметр. Дважды щелкните ЛК мыши в поле привязка и выберите «Реальное значение» канала «Канал_1»;

15. Сохраните проект в отдельной папке. Выполните команду «Сохранить для MPB». В результате в папке проекта появятся две поддиректории: RTM_1 и MicroRTM_2;

16. Скопируйте папки RTM_1 и MicroRTM_2 на соответствующие компьютеры;

17. Загрузите на компьютере 2 профайлер без поддержки отображения графических экранов оператора (rtmg32.exe), откройте в нем файл *.rtp из скопированной папки MicroRTM_2 и запустите пересчет каналов;

18. На компьютере 1 загрузите профайлер с поддержкой отображения графических экранов оператора (rtc.exe), откройте в нем файл *.dbb из папки RTM_1 и запустите пересчет каналов. В рабочем поле оболочки отобразится экран оператора.



а)

б)

а) – компьютер диспетчера; б)- компьютер, имитирующий работу RTU

Рисунок 9.1 – Экранные формы программ моделирования работы АРМ диспетчера

Пример результата выполнения представлен на рисунке 9.1

9.4 Контрольные вопросы

1. Поясните понятие SCADA-системы.
2. Назовите достоинства способа разработки АСУТП с использованием инструментальных средств SCADA-систем.
3. Назовите основные функции SCADA-систем.
4. Назовите основные элементы среды разработки системы «Trace Mode 6.0»
5. Поясните понятие «канал в системе «Trace Mode 6.0»».

10 Литература, рекомендуемая для изучения тем

10.1 Основная литература

1. Аралбаев, Т.З. Построение адаптивных систем мониторинга и диагностирования сложных промышленных объектов на основе принципов самоорганизации/Т.З. Аралбаев. – Уфа: Гилем, 2003.-248 с.
2. Аралбаев, Т.З. Структурно-параметрический и структурно-топологический синтез распределенных систем контроля и управления объектами нефтегазодобычи/Т.З. Аралбаев, Галимов Р.Р. - Уфа: Гилем, 2010.-144 с.
3. Бройдо, В.Л. Вычислительные системы, сети и телекоммуникации/В.Л. Бройдо. –М.: Спб.: Питер, 2006. –688 с.
4. Гамбаров, Г.М.Статистическое моделирование и прогнозирование/ Под ред. А.Г. Гранберга.- Н.М. Журавель, Ю.Г. Королев и др. - М.: Финансы и статистика, 1990. -384 с.
5. Дьяконов, В.П. MathCAD – 8/2000:Справочник/ В.П. Дьяконов.– С-Петербург: Питер, 2001. – 592 с.
6. Смирнова, Г.Н. Проектирование экономических информационных систем/ Г.Н. Смирнова, Ю.Ф. Тельнов. –М.: Финансы и статистика, 2003. -512 с.

10.2 Дополнительная литература

1. Социально-экономическое положение Оренбургской области/ Оренбургский областной комитет государственной статистики. - Оренбург: Госкомстат России, 1999. – 227 с.
2. Пьявченко, Т.А. Проектирование АСУТП в SCADA-системе: учебное пособие / А.Т. Пьявченко. – Таганрог, 2007. – 78 с.
3. Trace mode. Материал из Википедии – свободной энциклопедии. [Электронный ресурс]. - Режим доступа: http://ru.wikipedia.org/wiki/Trace_mode
4. Горюнов, А.Г. Архитектура микроконтроллера intel 8051: учеб. пособие./ А.Г. Горюнов, С.Н. Ливенцов. -Томск: Изд-во ТПУ, 2005. - 86 с.

Приложение А

(обязательное)

Листинг программы

```
Name Lr2
; указание адресов регистров специальных назначений
SV EQU    7h
DPP EQU   84h
T2CONEQU  0xC8
TF2 BIT   0xCF;
EXF2 BIT  0xCE;
RCLK BIT  0xCD;
TCLK BIT  0xCC;
XEN BIT   0xCB;
TR2 BIT   0xCA;
CNT2 BIT  0xC9;
CAP2 BIT  0xC8;
RCAP2L    EQU  0xCA;
RCAP2H EQU  0xCB;
ET2 BIT   0xAD;
CSEG AT 0h
jmp START ; переход на начало программы
; ниже приведенные директива и команда используются только при отладке, так как по адресу 02Bh
; не может быть размещена пользовательская программа в стенде
ORG 02Bh; по адресу 02Bh должен находиться обработчик прерывания таймера 2
jmp Timer2; переход на обработчик, который размещен в пользовательской области памяти
ORG 202Bh
jmp Timer2
ORG 2100h
START:
    mov r3,#0; инициализация значения счетчика
    mov r7,#0
    CLR RCLK ;устанавливаем 16-битный режим
    CLR TCLK ; с автоперезагрузкой для таймера 2
    CLR CAP2
    CLR TR2
    mov Rcap2h,#0x3c ; помещаем значение 15536
    MOV RCAP2L,#0xB0 ; в регистры RCAP2h и RCAP2l
    SETB EA ;разрешаем прерывания от таймера2
    SETB ET2
    SETB TR2 ; запускаем таймер2
cycle:
    inc r3; увеличиваем значение регистра r3 и
    Call WriteSvet; выводим на светодиоды
cycle2:CJNE r7,#20,cycle2;ожидаем пока таймер 2 переполнится 20 раз, что будет соответствовать 1 с
    mov r7,#0
    jmp cycle
Timer2: ; обработчик прерывания таймера 2
    inc r7
    CLR TF2; необходимо сбрасывать флаг переполнения TF2
    reti
WriteSvet: ; процедура
    mov r0,DPP
    mov DPP,#8
    mov r1,#sv
    mov a,r3
    mov dph,#0
    movx @r1,a
    mov DPP,r0
    Ret
END
```