

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Оренбургский государственный университет»

Кафедра вычислительной техники и защиты информации

МОДЕЛИРОВАНИЕ ПРОЦЕССОВ МОНИТОРИНГА МОБИЛЬНЫХ ОБЪЕКТОВ ИНФОРМАТИЗАЦИИ

Методические указания

Рекомендовано к изданию редакционно-издательским советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» для обучающихся по образовательным программам высшего образования по направлениям подготовки 09.03.01 Информатика и вычислительная техника и 10.03.01 Информационная безопасность

Оренбург
2019

УДК 004.056:519.85(076.5)
ББК 32.972.5-018.2я7
М 74

Рецензент - кандидат технических наук, доцент А.Ю. Кручинин
Авторы: Аралбаев Т.З., Галимов Р.Р., Абрамова Т.В., Каменева Е.В.

М 74 Моделирование процессов мониторинга мобильных объектов информатизации: методические указания / Т.З. Аралбаев, Р.Р. Галимов, Т.В. Абрамова, Е.В. Каменева; Оренбургский гос. ун-т. - Оренбург: ОГУ, 2019. – 77 с.

Методические указания содержат материалы для проведения 9 лабораторных работ. Каждая работа включает теоретическое изложение материала, постановку задачи, порядок выполнения и контрольные вопросы для самоподготовки.

Методические указания рекомендованы преподавателям как вспомогательный материал в организации и проведении занятий, а также студентам по направлениям подготовки 09.03.01 Информатика и вычислительная техника, профиль «Вычислительные машины, комплексы, системы и сети» и 10.03.01 Информационная безопасность, профиль «Комплексная защита объектов информатизации» - для аудиторного и самостоятельного освоения лабораторного курса дисциплин: «Проектирование вычислительных систем», «Проектирование вычислительных систем в защищенном исполнении». Методические указания также могут быть использованы при выполнении выпускных квалификационных работ студентами этих профилей.

УДК 004.056:519.85(076.5)
ББК 32.972.5-018.2я7

© Аралбаев Т.З.,
Галимов Р.Р.,
Абрамова Т.В.,
Каменева Е.В., 2019
© ОГУ, 2019

Содержание

Обозначения и сокращения	4
Введение	5
1 Лабораторная работа № 1. (4 часа). Идентификация положения мобильного объекта в трехмерном пространстве с учетом погрешностей навигационной аппаратуры на базе табличного процессора.....	7
2 Лабораторная работа № 2 (4 часа). Моделирование мониторинга перемещения мобильного объекта в трехмерном пространстве в среде табличного процессора с макросами.....	14
3 Лабораторная работа № 3 (4 часа). Моделирование процесса мониторинга мобильного объекта информатизации в среде табличного процессора с использованием среды разработки Lazarus	25
4 Лабораторная работа № 4 (4 часа). Идентификация положения мобильного объекта с использованием навигационной аппаратуры и табличного процессора....	33
5 Лабораторная работа № 5 (4 часа). Дистанционное изучение беспилотного мобильного объекта	37
6 Лабораторная работа № 6 (4 часа). Исследование сетевого трафика беспилотного мобильного объекта	45
7 Лабораторная работа № 7 (4 часа). Метод ранжирования рисков на основе ассоциативного подхода	53
8 Лабораторная работа № 8. Метод оценки положения МО в трехмерном пространстве с использованием микроконтроллера ADuC812 и теории нечеткой логики	58
9 Лабораторная работа № 9 (4 часа). Контроль команд оператора в системе управления объектом.	68
10 Литература, рекомендуемая для изучения тем	75
Приложение А.....	77

Обозначения и сокращения

АП – ассоциативная память

АМП – ассоциативно-мажоритарный подход

ГСЧ – генератор случайных чисел

БМО – беспилотный мобильный объект

ИБ – информационная безопасность

ИВТ – информатика и вычислительная техника

КЗОИ – комплексные системы защиты объектов информатизации

МК – микроконтроллер

МО – мобильный объект

МОИ – мобильный объект информатизации

ПК – персональный компьютер

ПУ – пульт управления

СТ – сетевой трафик

НС – навигационная система

РИВС – распределенная вычислительная система

GPS – Global Positioning System

OLE (Object Linking and Embedding) – технология связывания и внедрения объектов в другие документы и объекты, разработанная корпорацией Майкрософт

VBA (Visual Basic Application) – язык программирования

NMEA (National Marine Electronics Association) – стандарт определяющий текстовый протокол связи навигационного оборудования

UART (Universal Asynchronous Receiver-Transmitter) – последовательный интерфейс передачи данных

Введение

Настоящие методические указания к лабораторным работам предназначены для получения практических навыков студентами направлений «Информатика и вычислительная техника» (ИВТ) и «Информационная безопасность» (ИБ) по профилям подготовки - «Вычислительные машины, комплексы, системы и сети» и «Комплексная защита объектов информатизации» - для аудиторного и самостоятельного освоения лабораторного курса дисциплин: «Моделирование информационных процессов», «Проектирование вычислительных систем», «Проектирование вычислительных систем в защищенном исполнении», «Основы теории распознавания образов», «Системы распознавания образов».

Методические указания также будут полезны студентам, обучающимся по магистерским программам по направлениям «ИВТ» и «ИБ» при изучении дисциплины «Методы распознавания и прогнозирования процессов в распределенных вычислительных системах».

Лабораторный курс содержит девять работ, рассчитанных на 36 часов аудиторных занятий. Предлагаемые задания охватывают основные разделы рабочих программ, связанные с изучением технологии и организации моделирования контролируемых объектов и объектов защиты.

Специфика лабораторного курса определена тем, что в качестве исследуемых рассматриваются мобильные объекты (МО), изменяющие свое положение в трехмерном пространстве: беспилотные летательные аппараты, различные объекты из класса Интернет-вещей.

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;

– при использовании в процессе занятий МО необходимо соблюдать соответствующие правила техники безопасности;

– программное обеспечение для поддержки указаний представлено в прилагаемой к указаниям дискете.

Лабораторный курс может быть освоен на индивидуальном компьютере со средними техническими характеристиками. Обязательным для полной реализации курса является наличие комплекта офисных приложений, 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 указанных дисциплин.

1 Лабораторная работа № 1. (4 часа). Идентификация положения мобильного объекта в трехмерном пространстве с учетом погрешностей навигационной аппаратуры на базе табличного процессора

Цель работы: освоить процедуру автоматизированного мониторинга мобильного объекта в трехмерном пространстве с учетом погрешностей навигационной аппаратуры на базе табличного процессора.

1.1 Теоретическая часть

Актуальность темы работы определяется необходимостью подготовки и предоставления методического материала студентам, изучающим вопросы защиты мобильных объектов (МО) информатизации на базе доступных и иллюстративных средств вычислительной техники. В качестве МО в работе использован квадрокоптер типа Drone Parrot Bebop 2 [3].

Для достижения цели в работе представлены: графическое и математическое описания процесса мониторинга МО, порядок построения имитационной модели, алгоритм изучения имитационной модели и экранная форма полученных результатов.

Под мониторингом понимается процесс наблюдения за положением МО в трехмерном пространстве.

Имитационная модель – это система математической, алгоритмической и программной моделей, позволяющих представлять и исследовать в компьютерной системе реальные процессы. Основное требование к моделям – это адекватность (соответствие) представления реальным процессам.

Графическая модель трехмерного пространства перемещения МО с указанием зон неопределенности, обусловленными погрешностями навигационной аппаратуры представлена на рисунке 1.1. Область перемещения МО представлена в виде вложенных параллелепипедов, границы каждого из которых определяют зоны нормального перемещения МО (**S1**), допустимого перемещения (**S2**) и критического

перемещения (**S3**). Границы зон, определены координатами: z_1 - z_6 , находящимися в зонах неопределенности, выделенных прямоугольниками, обусловленными погрешностью навигационной аппаратуры. При получении координат, попадаемых в зоны z в соответствии с допущениями в моделировании, считается, что МО с одинаковой вероятностью может находиться в любой из граничных зон. Следует отметить, что в зависимости от используемой аппаратуры и условий мониторинга погрешность измерения координат может находиться в диапазоне от нескольких сантиметров до нескольких десятков метров.

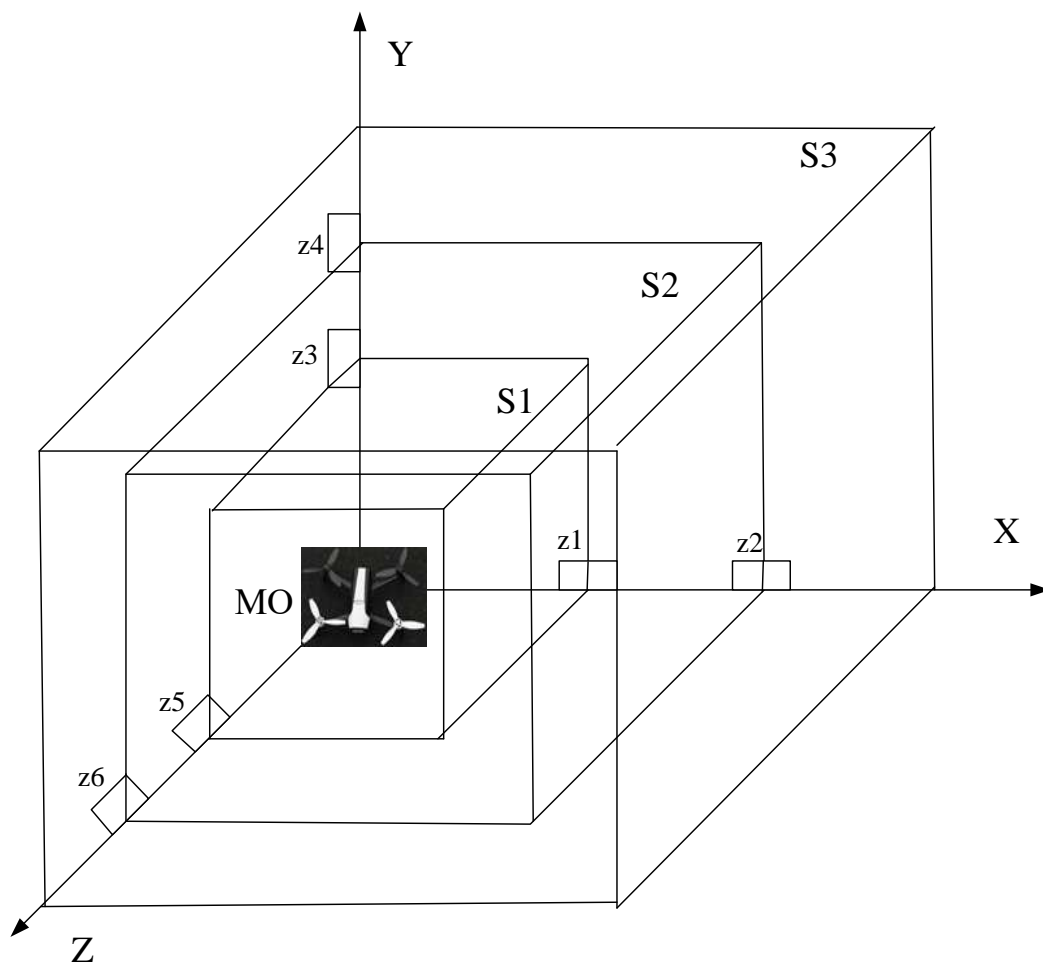


Рисунок 1.1 – Трехмерная графическая модель пространства перемещения мобильного объекта

Для описания задачи идентификации зоны МО при построении математической модели использованы следующие условные обозначения [1]:

- $S = \{S_1, S_2, S_3\}$ – множество контролируемых зон МО;
- q^x – неизвестный образ зоны, подлежащий распознаванию;
- S^* – образ зоны, к которому отнесен q^x ;
- $P = \{X, Y, Z\}$ – множество признаков распознавания образов;
- $\langle p_i \rangle$ – зарегистрированное значение i -го признака, $i = 1, 3$;
- $\langle P^x \rangle$ – вектор зарегистрированных значений признаков q^x ;
- $D = \{DX_1, DY, DZ\}$ – множество диапазонов изменения признаков, состоящее из множеств поддиапазонов признаков для каждого образа из множества S ;
- $V\{q^x, S_j\}$ – мера близости между q^x и j -ым образом из множества S , $j = 1, 3$;
- $v_{ij}\{\langle p_i \rangle, S_j\}$ – частный параметр (коэффициент) ассоциативности значения $\langle p_i \rangle$ признака p_i из множества P^x для образа S_j ;
- $\Phi\{\langle P^x \rangle, S_j\}$ – разделяющая функция для вычисления меры близости $V\{q^x, S_j\}$;
- $W_j\{\langle P^x \rangle, D_j\}$ – вектор коэффициентов оценок ассоциативности, характеризующих состояние принадлежности зарегистрированных значений q^x диапазонам значений признаков j -го образа из множества S , $j = 1, 3$;

Модель распознавания q^x имеет следующий вид:

$$V\{q^x, S_j\} = \Phi\{\langle P^x \rangle, S_j\}, j = 1, 3; \quad (1.1)$$

$$\Phi\{\langle P^x \rangle, S_j\} = \sum_{i=1}^M v_{ij}\{\langle p_i^x \rangle, S_j\}, i = 1, 3; \quad (1.2)$$

$$v_{ij}\{\langle p_i^x \rangle, S_j\} = \begin{cases} 1, & \text{если } \langle p_i^x \rangle \in D_{ij}; \\ 0, & \text{если } \langle p_i^x \rangle \notin D_{ij}; \end{cases} \quad (1.3)$$

$$q^x \in S^* \in S : V\{q^x, S^*\} \equiv \max V\{q^x, S_j\}, S_j \in S. \quad (1.4)$$

Выражение (1.4) представляет собой правило отнесения q^x к одному из образов множества S по принципу простого большинства с использованием меры Хемминга, либо по принципу превышения суммы ассоциаций более половины признаков.

В зависимости от данных навигационной аппаратуры и принятой системы кодирования состояний положение МО может быть описано кодовыми сигнатурами, представленными в таблице 1.1. В таблице 1.1 темным цветом выделены коды несуществующих состояний положения мобильного объекта.

Таблица 1.1 – Таблица кодов состояний положения мобильного объекта с учетом погрешностей навигационной аппаратуры

№ п. п.	X			Y			Z		
	S1	S2	S3	S1	S2	S3	S1	S2	S3
1	0	0	0	0	0	0	0	0	0
2	A			A			A		
	0	0	1	0	0	1	0	0	1
3	B			B			B		
	0	1	0	0	1	0	0	1	0
4	C			C			C		
	0	1	1	0	1	1	0	1	1
5	D			D			D		
	1	0	0	1	0	0	1	0	0
6	1	0	1	1	0	1	1	0	1
7	E			E			E		
	1	1	0	1	1	0	1	1	0
8	1	1	1	1	1	1	1	1	1

В таблице 1.2 представлены характеристики некоторых сигнатур положения МО.

Таблица 1.2 – Таблица соответствия кодов и положения мобильного объекта

№ п.п.	Совокупные коды координат			Положение мобильного объекта
1	2			3
1	A	A	A	Критическое положение МО по координатам X, Y, Z
2	A	A	B	Критическое положение МО по координатам X, Y, а по Z – в допустимой зоне
3	B	B	B	Допустимое (граничное) положение МО по всем координатам

Продолжение таблицы 1.2

1	2			3
4	A	A	E	Критическое положение МО по координатам X, Y, а по Z в нормально-допустимой зоне
5	C	C	C	Положение МО по всем координатам в допустимо-критической зоне
6	B	C	E	Положение МО по всем координате Z в допустимо-критической зоне
7	D	D	D	Положение МО по всем координате в нормальной зоне
8	E	E	D	Положение МО по всем координатам X и Y в нормально допустимой зоне
9	E	E	E	Положение МО по всем координатам в допустимой зоне

1.2 Порядок выполнения работы

Порядок построения и исследования имитационной модели с использованием табличного процессора Excel представлен на экранной форме имитационной модели (рисунок 1.2) и имеет нижеперечисленные пункты.

1. Запустить программу табличного процессора «Мониторинг положения МО». Программа находится на прилагаемой к курсу дискете;

2. В соответствии с данными варианта формируется экранное пространство табличного процессора для имитационного моделирования процесса мониторинга МО. При этом задаются различными цветами области диапазонов координат перемещения МО, в частности для зоны **S1, S2, S3**;

3. С использованием генератора случайных чисел (ГСЧ) в ячейках, выделенных желтым цветом, задаются координаты по осям **X, Y, и Z**;

4. Для получения фактических координат производится их пересчет в ячейках, находящихся левее данных ГСЧ, при этом общая величина двух соседних диапазонов перемножается на коэффициент, выданный генератором;

5. В соответствии с математическими выражениями программируется таблица для идентификации состояния **S***;

6. Провести анализ корректности заполнения результатных данных верхней таблицы экранной формы;

7. Провести анализ корректности результатных данных нижней таблицы.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Xn	Xd	Xk		Yn	Yd	Yk		Zn	Zd	Zk							
2																		
3	S1	S2	S3		S1	S2	S3		S1	S2	S3							
4																		
5	7,331	0,123			10,82	0,284			21,92	0,597			x1	x2	x3	C	S	
6	21,12	0,634			23,55	0,814			21,24	0,575			1	1	1	3	s1	
7	10,01	0,223			5,501	0,063			14,81	0,36			0	1	0	1	s2	
8													0	1	0	1	s2	
9													1	0	1	2	s3	
10																		
11													y1	y2	y3	C		
12													1	0	1	2	s1	
13													0	1	0	1	s2	
14																		
15						10,82							1	0	0	1	s2	
16						23,55							1	1	1	3	s3	
17						5,501												
18		7,331								21,92								
19		21,12								21,24			z1	z2	z3	C		
20		10,01								14,81			1	1	1	3	s1	
21													1	0	0	1	s2	
22													0	0	1	1	s2	
23													1	1	1	3	s3	
24																		
25							25,88	0,429					№ замера	S1 или S2	№ замера	S2 или S3		
26							37,22	0,807					по оси X	по оси X				
27							30,88	0,596					1	1	1	3		
28																		
29													№ замера	S1 или S2	№ замера	S2 или S3		
30			41,78	0,951									по оси Y	по оси Y				
31			17,07	0,003									1	1	1	3		
32			36,73	0,759														
33													№ замера	S1 или S2	№ замера	S2 или S3		
34													по оси Z	по оси Z				
35													1	1	1	3		
36																		
37																		
38																		
39																		
40																		
41																		
42																		
43																		
44																		

Рисунок 1.2 – Экранная форма процесса имитационного моделирования перемещения МО в трехмерном пространстве с использованием макросов табличного процессора

1.3 Задание на лабораторную работу

1. В соответствии с данными варианта работы построить имитационную модель в среде табличного процессора Excel.
2. Определить положение объекта для трех наборов координат, заданных генератором случайных чисел.
3. Устно ответить на контрольные вопросы.
4. Составить отчет по лабораторной работе, в котором указать:
 - тему и цель работы;
 - экранную форму имитационной модели;
 - вывод по проделанной работе в соответствии с поставленной целью.

1.4 Варианты заданий

Варианты заданий для выполнения лабораторной работы представлены в таблице 1.3.

Таблица 1.3 – Варианты заданий для выполнения лабораторной работы

Номер варианта	Диапазоны изменения координат								
	Зона S1			Зона S2			Зона S3		
	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ
1	6-25	6-22	6-26	19-33	15-30	17-36	30-44	26-44	32-44
2	5-26	5-23	7-25	20-32	19-34	21-33	28-44	30-44	29-44
3	6-22	8-24	9-24	18-32	20-30	20-30	30-44	28-44	28-44
4	5-24	5-22	5-26	18-32	14-28	16-36	28-44	28-44	28-44
5	5-26	7-24	8-28	20-32	20-30	24-36	22-44	26-44	3-44
6	6-25	6-20	6-26	19-33	15-30	17-36	30-44	26-44	32-44
7	5-24	5-20	5-26	18-32	14-28	16-36	28-44	28-44	28-44
8	5-26	5-24	7-25	20-32	19-34	21-33	28-44	30-44	29-44
9	6-24	8-26	9-24	18-32	20-30	20-30	30-44	28-44	28-44
10	5-22	5-23	7-25	18-32	19-34	21-33	28-44	30-44	29-44

1.5 Контрольные вопросы

- 1 Дайте определение имитационной модели.
- 2 Дайте определение процессу мониторинга объектов.
- 3 Чем объясняется область пересечения координат зон.
- 4 Что означает мажоритарный принцип принятия решений.
- 5 В чем отличие дрона от квадрокоптера.
- 6 Приведите примеры областей применения дрона в промышленности.
- 7 Какие требования предъявляются к имитационной модели объекта.
- 8 Укажите достоинства и недостатки имитационного моделирования в среде табличного процессора.
- 9 Объясните математическую модель определения положения МО.

2 Лабораторная работа № 2 (4 часа). Моделирование мониторинга перемещения мобильного объекта в трехмерном пространстве в среде табличного процессора с макросами

Цель работы: получить навыки по использованию макросов для автоматизации проведения вычислительных экспериментов.

2.1 Теоретическая часть

Разработанная в табличном процессоре Excel имитационная модель на основе формул, представленная в работе №1, позволяет получить результаты только для текущих данных. Для исследования модели в динамике для ряда последовательных тактов применение только механизма встроенных формул является недостаточным, поскольку необходимо выполнить ряд действий: сохранять данные предыдущего цикла, задавать текущие данные для каждого нового шага, каждый раз осуществлять расчеты и выводить все результаты в виде графиков. Одним из решений задачи устранения этого недостатка является использование макросов, созданных при помощи языка Visual Basic Application (VBA) [4].

Язык VBA встроен в приложения пакета Microsoft Office, такие как Word, Excel, Power Point. Visual Basic Application прост в изучении, разработанный код не требует компиляции и сохраняется непосредственно в самом документе. Несмотря на то, что разработанные на VBA программы не характеризуются высоким быстродействием в отличие от других языков программирования, данное средство полезно для специалистов и инженеров для автоматизации ежедневных задач при минимальных временных затратах на разработку.

VBA-программы часто называют макросами. Самый простой способ создать макрос – использовать макрокодер, который можно рассматривать как магнитофон, записывающий последовательность действий пользователя. Далее запись сохраняется и может быть «проиграна» по требованию пользователя.

Ниже приведен пример использования макрокодера. Постановка задачи: пользователю часто требуется сделать выделенный текст в формате «полужирный»

и «курсив» в Excel. Требуется автоматизировать изменение формата текста.

Решение. Запустите приложение Excel и сохраните документ с типом «Книга с поддержкой макросов» (рисунок 2.1).

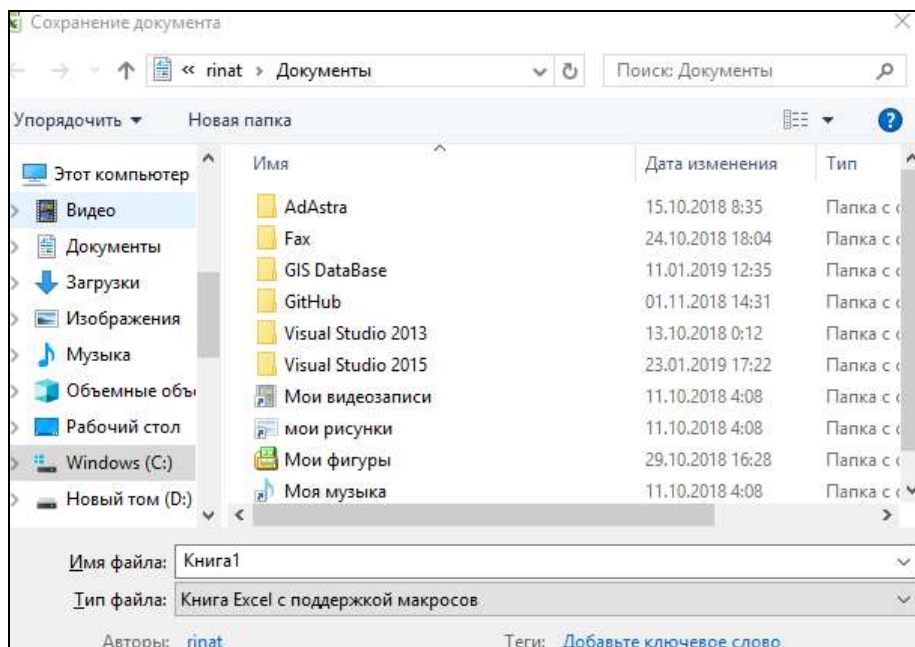


Рисунок 2.1 – Окно сохранения документа

Напишите произвольный текст в определенную ячейку документа, выделите её и выберите команду «Вид|Макросы|Запись» (рисунок 2.2).

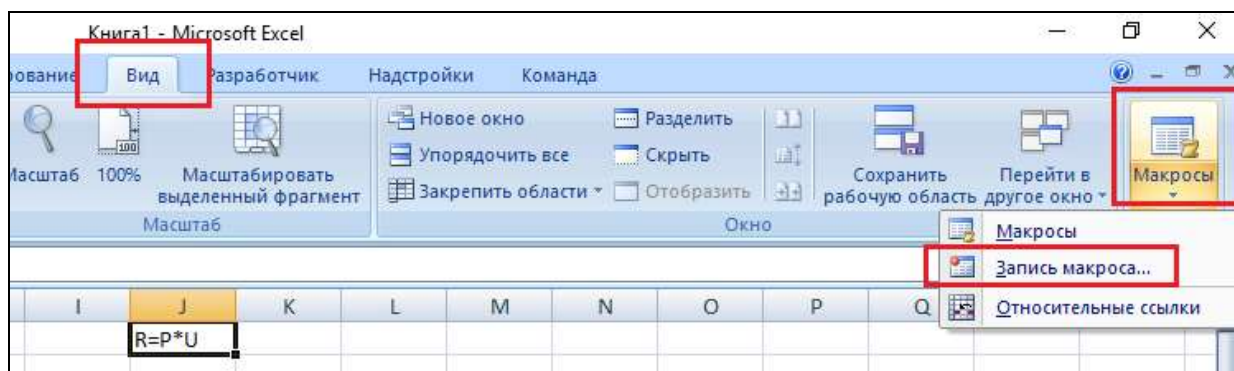


Рисунок 2.2 – Выбор команды запуска макрокодера

Далее задайте параметры макроса, в том числе и «горячие клавиши».

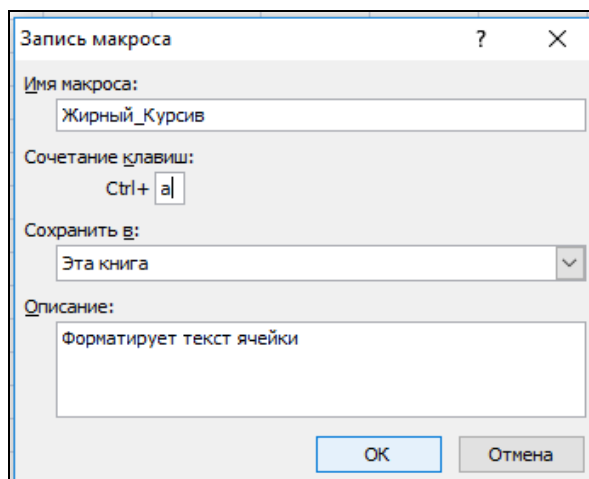


Рисунок 2.3 – Окно задания параметров макроса

После этого перейдите на вкладку «Главная» и измените формат шрифта выбранной ячейки на следующие: полужирный и курсив (эти действия будут «записаны»). Для остановки «записи» инструкций выберите команду «Вид| Макросы| Остановить запись» (рисунок 2.4).

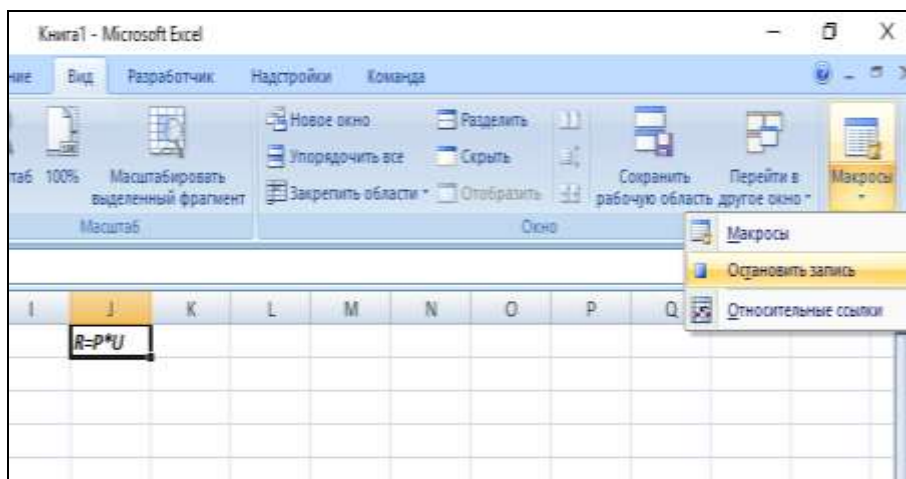


Рисунок 2.4 – Окно выбора команды «Остановить запись»

При успешном окончании «записи» разработанный макрос можно вызывать следующими способами:

- с помощью горячих клавиш (в примере Ctrl+a);
- выбрав команду «Вид| Макросы| Макросы». В появившемся окне выбираем

нужный макрос и нажимаем на кнопку «Выполнить» (рисунок 2.5).

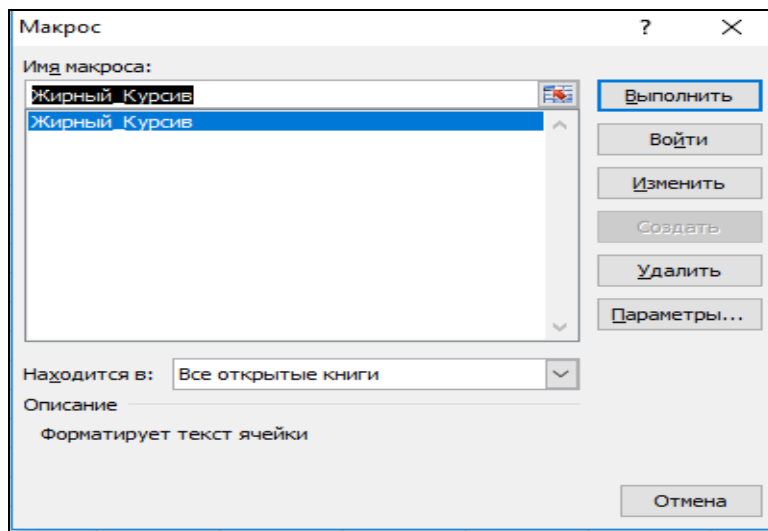


Рисунок 2.5 – Окно выбора макроса для выполнения

Несмотря на то, что непосредственно пользователь в данном случае не писал код, но макрос сохранился как последовательность команд на языке VBA. Код можно посмотреть, выбрав в окне, представленном на рисунке 2.5, команду изменить. На рисунке 2.6 представлен данный программный код.

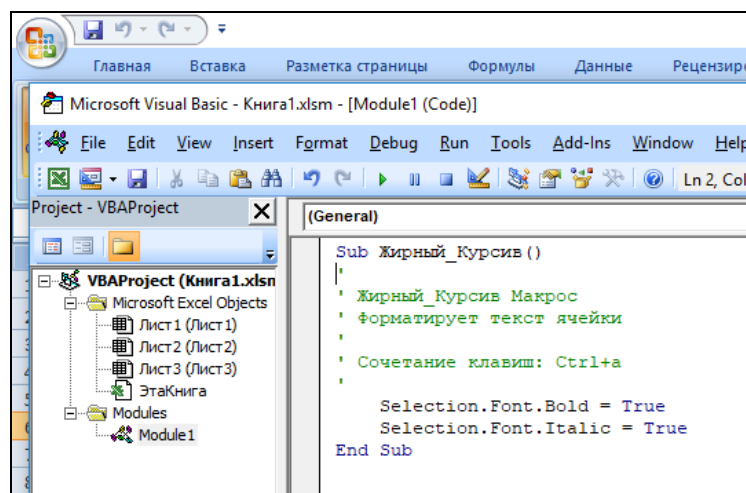


Рисунок 2.6 – Окно редактирование программного кода

Среда программирования Visual Basic Application имеет широкие возможности по разработке кода: редактирование, отладка (Debug) и выполнения (Run) программы. Кроме изменения свойств объектов приложений Microsoft Office (в примере изменяются параметры объекта Font), поддерживаются классические операторы языков программирования, например, условные операторы, операторы циклов. Также можно в коде использовать встроенные формулы. Более подробная информация представлена в справочной системе VBA.

Для формирования удобного пользовательского интерфейса вызова макросов VBA позволяет в документы добавлять графические элементы, например, кнопки. Рассмотрим пример размещения кнопки на листе документа Excel. Элементы управления размещены на вкладке «Разработчик». Для отображения данной вкладки нужно выбрать команду «Кнопка Office | Параметры Excel» (рисунок 2.7).

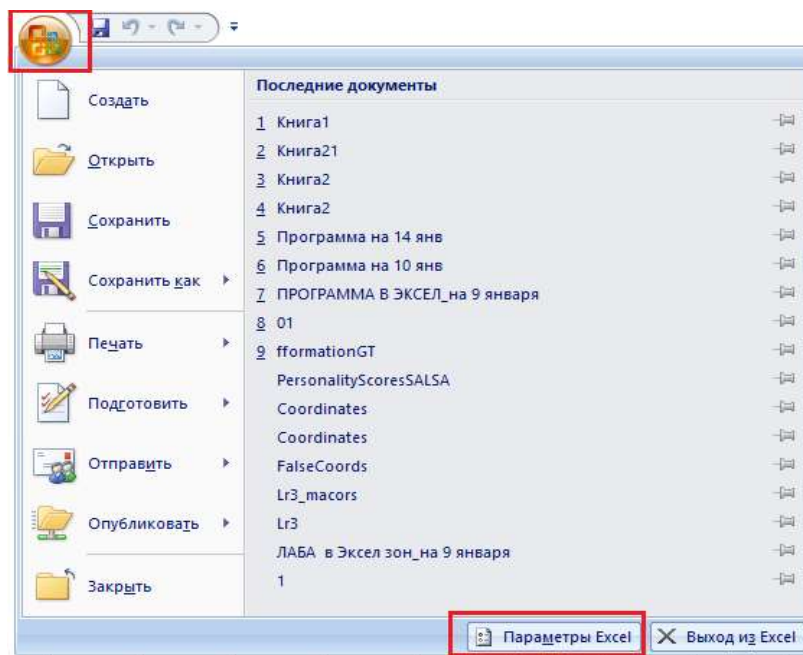


Рисунок 2.7 – Окно выбора параметров программы

В появившемся окне «Параметры Excel» в разделе «Основные» выставьте флажок «Показывать вкладку «Разработчик» на ленте» (рисунок 2.8).

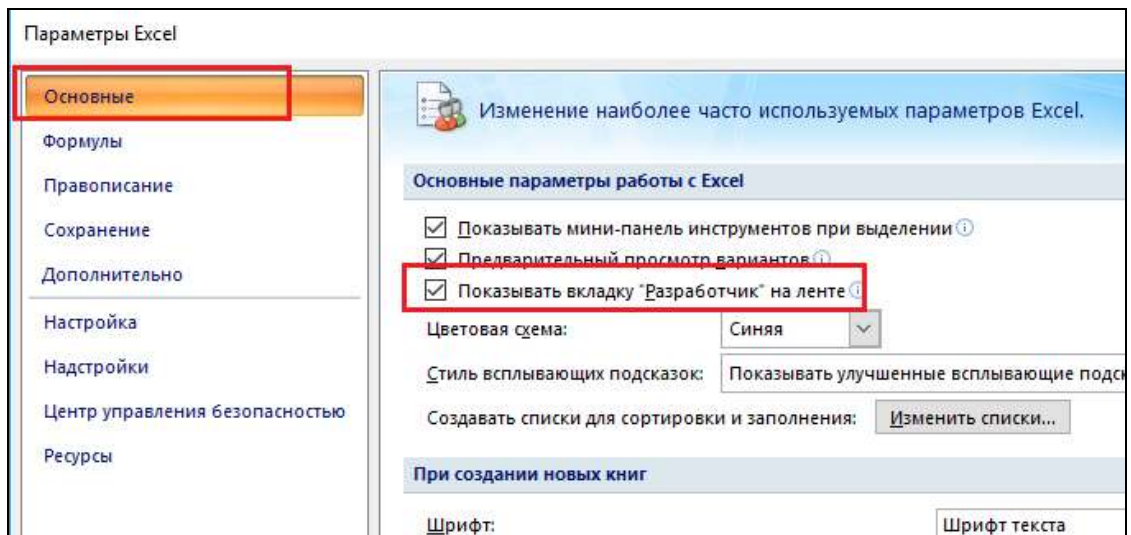


Рисунок 2.8 – Окно изменение параметров Excel

Далее переходим на вкладку «Разработчик| Вставить| Элементы управления формы| Кнопка» (рисунок 2.9).

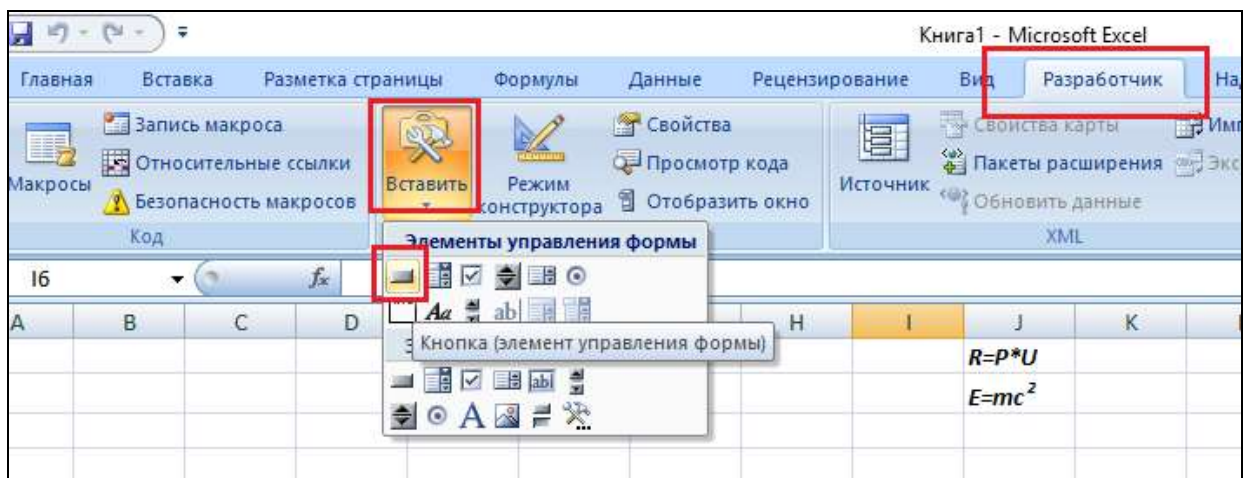


Рисунок 2.9 – Выбор элемента управления

При помощи мыши поместите элемент управления «Кнопка» на листе. Сразу же после размещения кнопки автоматически появляется окно, предлагающее создать макрос, который будет вызываться при щелчке мыши на данном элементе (рисунок 2.10).

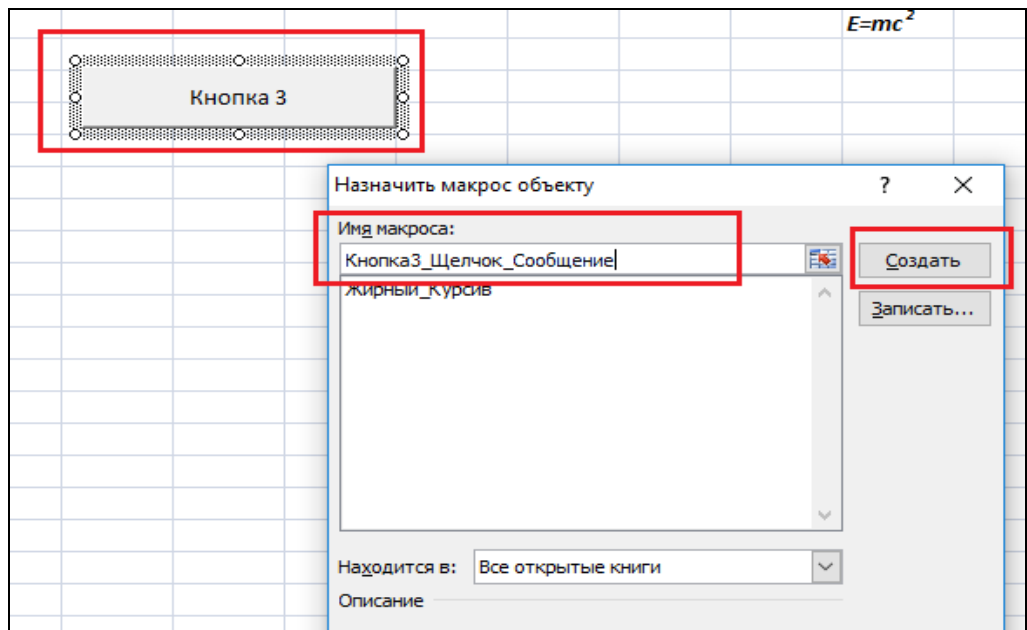


Рисунок 2.10 – Привязка макроса к событию кнопки

В результате окажемся в окне редакторе кода. Для примера напишем макрос, который заносит в ячейку A1 случайное число и выдает сообщение на экран. Код макроса представлен на рисунке 2.11.

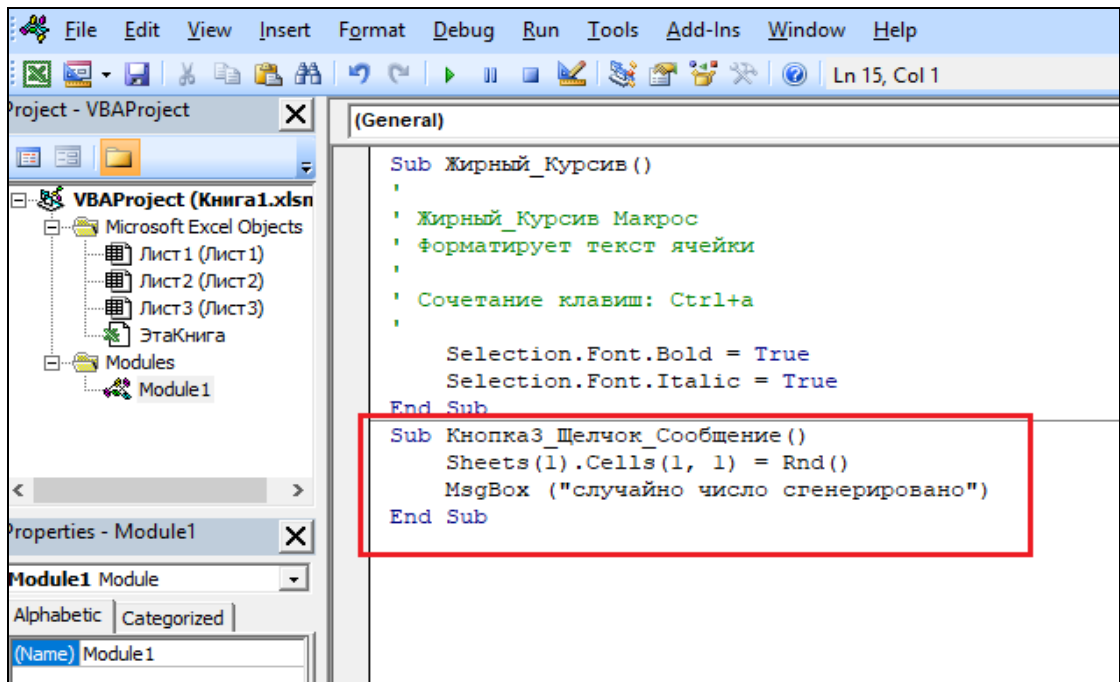


Рисунок 2.11 – Программный код

Сохраните код и проверьте, что данный макрос вызывается по щелчку мыши. Щелкнув правой кнопкой мыши на элементе управления, вызывается контекстное меню, в котором можно изменить свойства объекта. Например, изменить текст.

Таким образом, VBA представляет собой мощный инструмент для автоматизации и проведения «быстрых» расчетов, дополняющий стандартные возможности Excel. Но данные возможности часто также используют злоумышленники, для разработки и распространения вредоносного программного кода. Поэтому не нужно открывать документы с макросами, полученными из не доверенных источников.

2.2 Постановка задачи

1. Изучить теорию про язык программирования VBA и создание макросов.
2. Модифицировать имитационную модель при помощи макросов с целью генерации новых данных по нажатию кнопки и сохранения результатов в соответствующей номеру эксперимента строке. Предусмотреть проведение 10 экспериментов.

Возможный интерфейс новой модели представлен на рисунке 2.12. По нажатию кнопки «Следующий замер» генерируются новые входные данные координат по трем осям на основе случайных чисел. Данные таблицы «Текущие измерения» переносятся в соответствующие строки таблицы «Замеры» с учетом номера замера из ячейки R26. Графики строятся на основе данных таблицы «Замеры».

3. По результатам работы сделать выводы и составить отчет.
4. Ответить на контрольные вопросы.

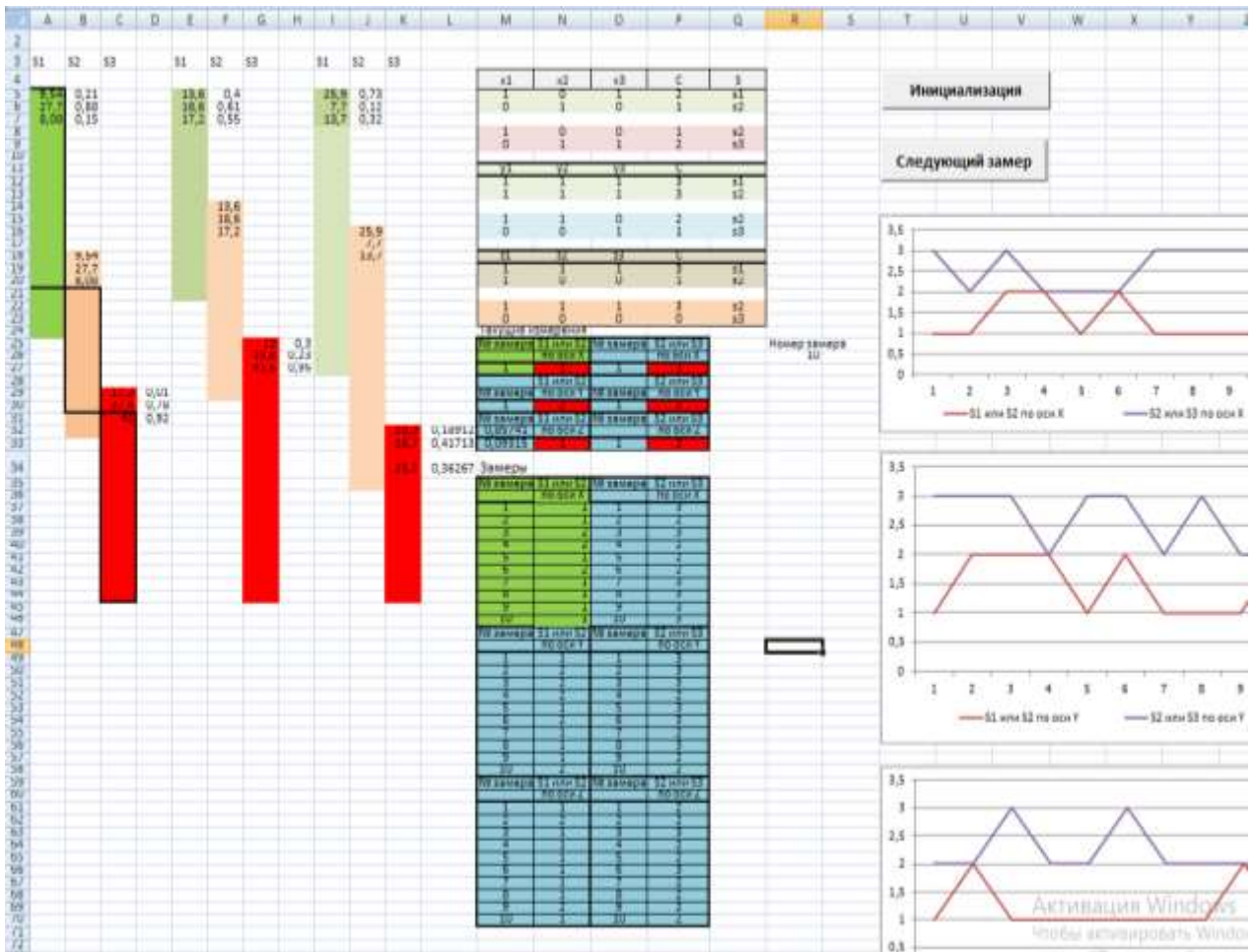


Рисунок 2.12 – Интерфейс имитационной модели на основе макросов

2.3 Порядок выполнения работы

1. Откройте документ Excel, сформированный в предыдущей работе. Сохраните его в формате «Книга с поддержкой макросов», выбрав команду «Сохранить как»;
2. Оформите таблицу «Замеры» и заполните значения только столбцов «№ замера».
3. Поместите кнопку на документе и назначьте ей макрос «Init», который будет обнулять значения таблицы «Замеры» и устанавливать значение номера счетчика в ячейке R26 равным нулю. Ниже приведена часть кода макроса:

```
Sub Кнопка2_Init()
    Worksheets(1).Range("R26") = 0
    Worksheets(1).Range("N37:N46").ClearContents
End Sub
```

```
Worksheets(1).Range("P37:P46").ClearContents
'.....
End Sub
```

4. Приведенный выше код очищает содержимое ячеек в диапазоне от N37 до N46 и от P37 до P46. Добавьте еще 4 команды, которые удаляют результаты предыдущего замера для оставшихся 4-х столбцов.

5. Измените надпись кнопки на текст «Инициализация».

6. Добавьте вторую кнопку и назначьте ей макрос «Next». Ниже приведена часть кода макроса:

```
Sub Кнопка1_Next()
    If Worksheets(1).Range("R26") >= 10 Then
        '-----выведите сообщение о превышении количества
        '--- заданных замеров
        Exit Sub
    End If
    Worksheets(1).Range("R26") = Worksheets(1).Range("R26") + 1
    Worksheets(1).Range("B5") = Rnd
    Worksheets(1).Range("B6") = Rnd
    Worksheets(1).Range("B7") = Rnd
    Worksheets(1).Range("F5") = Rnd
    Worksheets(1).Range("F6") = Rnd
    Worksheets(1).Range("F7") = Rnd
    Worksheets(1).Range("J5") = Rnd
    Worksheets(1).Range("J6") = Rnd
    Worksheets(1).Range("J7") = Rnd
    Worksheets(1).Range("D29") = Rnd
    Worksheets(1).Range("D30") = Rnd
    Worksheets(1).Range("D31") = Rnd
    Worksheets(1).Range("H25") = Rnd
    Worksheets(1).Range("H26") = Rnd
    Worksheets(1).Range("H27") = Rnd
    Worksheets(1).Range("L32") = Rnd
    Worksheets(1).Range("L33") = Rnd
    Worksheets(1).Range("L34") = Rnd
End Sub
```

```

Dim ind As Integer
ind = Worksheets(1).Range("R26")
Worksheets(1).Cells(37 + ind - 1, 14) = Worksheets(1).Cells(27, 14)
Worksheets(1).Cells(37 + ind - 1, 16) = Worksheets(1).Cells(27, 16)
' -----
End Sub

```

7. Добавьте возможность вывода сообщения о превышении допустимого количества замеров в блоке условия при помощи команды `msgbox`.

8. В приведенном выше коде в конце приведены команды заполнения столбцов состояний *S* по оси *X*. Аналогично заполните состояния для осей *Y* и *Z* (еще 4 столбца).

9. Добавьте 3 рисунка с графиками для двух состояний соответственно по каждой оси: *X*, *Y* и *Z*.

10. Сделайте выводы по работе и составьте отчет по работе.

2.4 Контрольные вопросы

1 Определите недостатки имитационной модели, представленной в 1-ой работе, с точки зрения получения динамических характеристик.

2 Что такое макрос?

3 Определите достоинства языка программирования VBA.

4 Что такое макрокoder?

5 Какое назначение команды `rnd`?

6 Почему нельзя открывать документы с макросами, полученными из не доверенных источников?

3 Лабораторная работа № 3 (4 часа). Моделирование процесса мониторинга мобильного объекта информатизации в среде табличного процессора с использованием среды разработки Lazarus

Цель работы: освоить принципы взаимодействия табличного процессора Excel и программного средства на языке высокого уровня при моделировании процессов мониторинга мобильного объекта.

3.1 Теоретическая часть

Актуальность темы работы определяется необходимостью подготовки и предоставления методического материала студентам, изучающим вопросы защиты и моделирования поведения мобильных объектов (МО) информатизации на базе доступных прикладных и инструментальных средств операционной системы.

Для достижения цели необходимо: изучить принципы работы технологии OLE при управлении табличным процессором Excel, разработать программу управления табличным процессором Excel на языке высокого уровня, исследовать работу программы в среде Excel и выявить достоинства и недостатки работы программы.

Модель многокомпонентных объектов COM (Component Object Model) это одна из архитектур предоставления программных сервисов. Программы, созданные с использованием COM, обеспечивают доступ к своим сервисам с помощью одного или нескольких COM-объектов, создаваемых в клиентских приложениях и являющихся экземплярами классов этих программ, называемых серверами COM.

Технологии OLE и ActiveX фирмы Microsoft, созданные на базе COM-модели, это ещё один шаг к более совершенным, т. е. более надежным и эффективным, программам. OLE-технология обеспечивает создание составных документов связыванием и внедрением объектов, определенных в серверных приложениях. Технология ActiveX обеспечивает использование всех остальных сервисов, которые возможно создать на базе COM-модели.

Технология позволяет одному приложению управлять другим приложением, например, обеспечивать обмен данными между табличным процессором и приложениями, разработанными на языках высокого уровня. В данной лабораторной работе в качестве табличного процессора рассматривается MS Excel, в качестве среды разработки – открытая среда разработки программного обеспечения Lazarus [6].

Пример управления данными в среде Excel с помощью программы, написанной в Lazarus, приведен на рисунке 3.1.

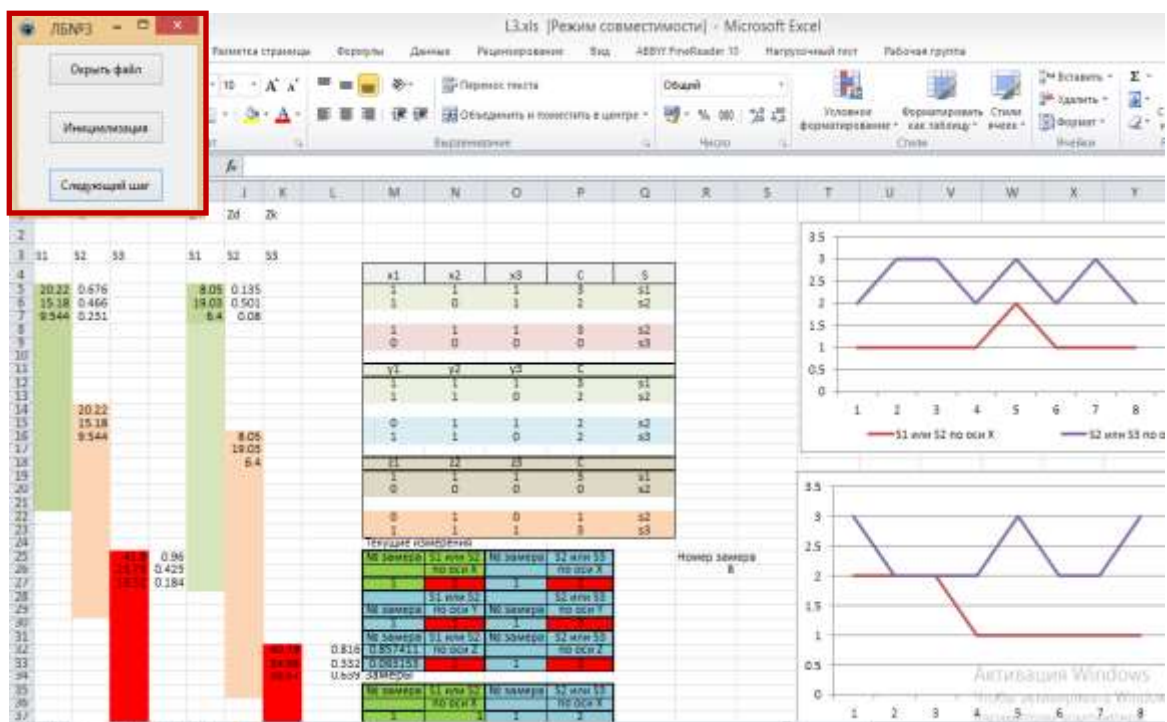


Рисунок 3.1 - Управление данными в среде Excel с помощью программы на языке высокого уровня

Для взаимодействия разрабатываемой программы с MS Excel в программе необходимо использовать модуль ComObj и объявить переменную для доступа к MS Excel вариантного типа следующим образом:

uses ComObj;

var Excel: Variant;

Инициализацию переменной Excel в простейшем случае можно осуществить так:

```
Excel := CreateOleObject('Excel.Application');
```

Создание новой книги:

```
Excel.Workbooks.Add;
```

Открытие существующей книги (где path - путь к файлу с расширением xls.):

```
Excel.Workbooks.Open[path];
```

Закрытие Excel:

```
Excel.ActiveWorkbook.Close;
```

```
Excel.Application.Quit;
```

Отображение или скрытие Excel на экране:

```
Excel.Visible := True;
```

```
Excel.Visible := False;
```

Доступ к ячейке в текущей книге Excel можно осуществить следующим образом:

```
Excel.Range['b2'] := 'Привет!'; // запись значения в ячейку
```

```
s := Excel.Range['b2']; // чтение значения из ячейки
```

где b2 - адрес ячейки.

Или используя стиль ссылок R1C1:

```
excel.Cells[2, 2] := 'Привет!';
```

где [2, 2] - координата ячейки.

Закрытие документа Excel и освобождение процесса:

```
XL.WorkBooks.Close;
```

```
XL.Quit;
```

При этом, ячейке Excel можно присваивать любое значение (символьное, целое, дробное, дата), Excel установит форматирование в ячейке, применяемое по умолчанию.

Достоинством управления данными табличного процессора с помощью программы, написанной на языке высокого уровня, являются:

– автоматизация процесса моделирования без использования макросов;

- широкие возможности работы с приложениями операционной системы, предоставляемые языками высокого уровня;

- оперативность обработки данных.

Недостатки:

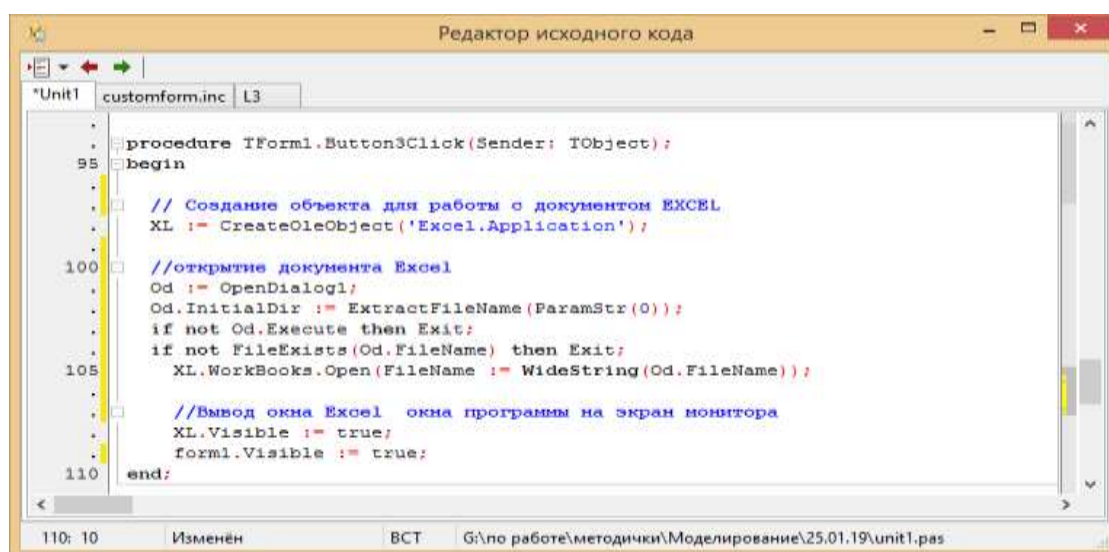
- необходимость знания языков программирования высокого уровня;

- технология OLE недоступна для управления табличными процессорами операционных систем Linux.

3.2 Порядок выполнения работы

На рисунках ниже приведен код программы, осуществляющий генерацию и запись случайных данных в ячейки Excel с последующим расчетом параметров процесса имитационного моделирования перемещения МО в трехмерном пространстве, описанных в лабораторной работе №1, и построением соответствующих графиков.

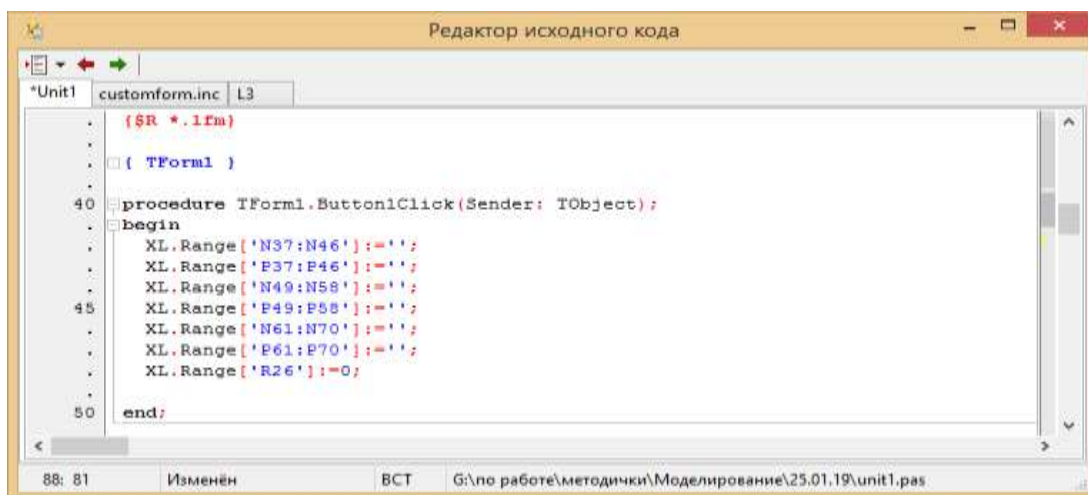
Разрабатываемая программа имеет кнопки «Открыть файл», «Инициализация» и «Следующий шаг», как показано на рисунке 3.1. При нажатии на кнопку «Открыть файл» при помощи технологии OLE осуществляется открытие в программе документа Excel. Код процедуры, осуществляющий открытие документа Excel в программе Lazarus представлен на рисунке 3.2.



```
Redатор исходного кода
Unit1 customform.inc L3
.
.
.
procedure TForm1.Button3Click(Sender: TObject);
95 begin
.
.
.
// Создание объекта для работы с документом EXCEL
.
.
.
XL := CreateOleObject('Excel.Application');
.
.
.
100 //открытие документа Excel
Od := OpenDialog1;
.
.
.
Od.InitialDir := ExtractFileName(ParamStr(0));
if not Od.Execute then Exit;
if not FileExists(Od.FileName) then Exit;
105 XL.WorkBooks.Open(FileName := WideString(Od.FileName));
.
.
.
//Вывод окна Excel окна программ на экран монитора
.
.
.
XL.Visible := true;
form1.Visible := true;
110 end;
.
.
.
110: 10 Изменён ВСТ G:\но работе\методички\Моделирование\25.01.19\unit1.pas
```

Рисунок 3.2 – Процедура открытия документа Excel в программе Lazarus

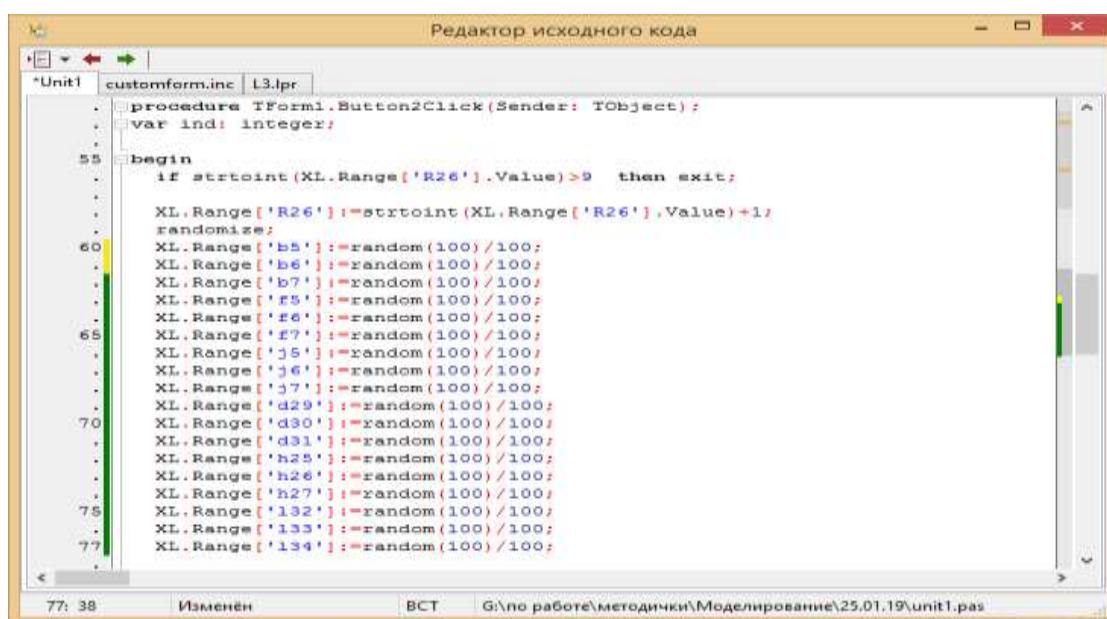
При нажатии кнопки «Инициализация» осуществляется подготовка документа Excel для генерации и обработки нового набора данных, как показано на рисунке 3.3:



```
Редатор исходного кода
*Unit1 customform.inc L3
{SR *.lrm}
[ TForm1 ]
40 procedure TForm1.Button1Click(Sender: TObject);
begin
  XL.Range['N37:N46']:= '';
  XL.Range['P37:P46']:= '';
  XL.Range['N49:N58']:= '';
  45 XL.Range['P49:P58']:= '';
  XL.Range['N61:N70']:= '';
  XL.Range['P61:P70']:= '';
  XL.Range['R26']:=0;
  50 end;
```

Рисунок 3.3 – Процедура очистки данных предыдущего эксперимента

При нажатии кнопки «Следующий шаг» в ячейках документа Excel B5-B7, F5-F7, J5-J7, D29-D31, H25-h27 и L32-L34 осуществляется запись сгенерированных программой случайных значений. Код соответствующей процедуры представлен на рисунке 3.4.



```
Редатор исходного кода
*Unit1 customform.inc L3.lpr
procedure TForm1.Button2Click(Sender: TObject);
var ind: integer;
55 begin
  if strtoint(XL.Range['R26'].Value)>9 then exit;
  XL.Range['R26']:=strtoint(XL.Range['R26'].Value)+1;
  randomize;
  60 XL.Range['b5']:=random(100)/100;
  XL.Range['b6']:=random(100)/100;
  XL.Range['b7']:=random(100)/100;
  XL.Range['f5']:=random(100)/100;
  XL.Range['f6']:=random(100)/100;
  65 XL.Range['f7']:=random(100)/100;
  XL.Range['j5']:=random(100)/100;
  XL.Range['j6']:=random(100)/100;
  XL.Range['j7']:=random(100)/100;
  XL.Range['d29']:=random(100)/100;
  70 XL.Range['d30']:=random(100)/100;
  XL.Range['d31']:=random(100)/100;
  XL.Range['h25']:=random(100)/100;
  XL.Range['h26']:=random(100)/100;
  XL.Range['h27']:=random(100)/100;
  75 XL.Range['l32']:=random(100)/100;
  XL.Range['l33']:=random(100)/100;
  77 XL.Range['l34']:=random(100)/100;
```

Рисунок 3.4 – Генерация и запись случайных значений в ячейки Excel

Далее в ячейки записываются расчетные данные для получения кодов состояний положения мобильного объекта и построения графиков, как показано на рисунке 3.5:

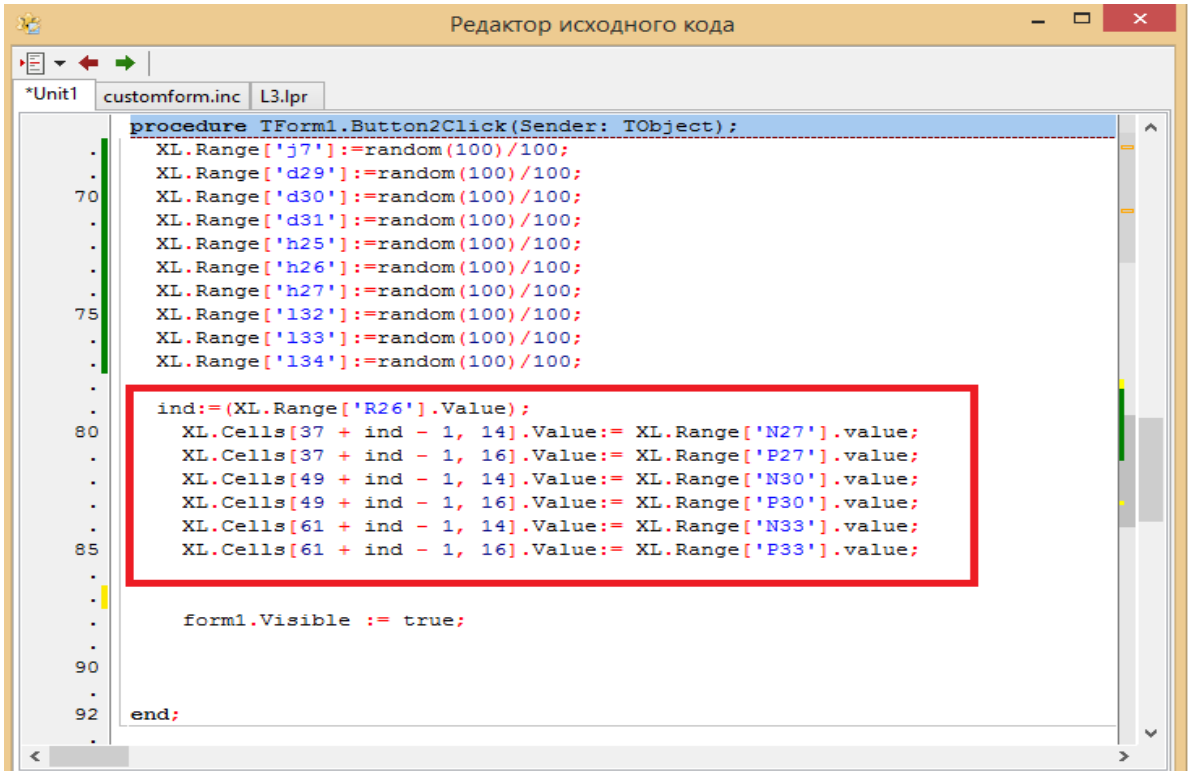


Рисунок 3.5 – Расчет и запись кодов состояний положения мобильного объекта

Для исключения ошибок в кодировке при чтении программой имени файла обрабатываемый файл Excel L3.xls необходимо разместить так, чтобы путь к файлу содержал только латинские буквы или цифры (например, D:\L3.xls или C:\Lazarus\L3.xls).

Если проект бы создан и скомпилирован в одной операционной системе Windows, а исполняемый файл запущен в другой, приложение может работать некорректно. Если при попытке запустить файл L3.exe приложение не открывается или работает некорректно, необходимо скомпилировать проект в используемой операционной системе. Для этого: открыть исходный код проекта (файл L3.lpr) и скомпилировать проект (нажать «Запуск – Компилировать»). Далее запустить исполняемый файл L3.exe, созданный в папке с исходным кодом программы.

3.3 Задание на лабораторную работу

1 В соответствии с данными варианта работы в среде программирования Lazarus разработать программу моделирования поведения мобильного объекта в среде табличного процессора Excel. Пример реализации основных процедур программы представлен на рисунках 3.2 – 3.5 пункта 3.2.

2 Используя предложенный преподавателем файл L3.xls, либо файл, полученный в результате выполнения лабораторной работы №1, исследовать работу программы в среде Excel и выявить достоинства и недостатки работы программы.

3 Устно ответить на контрольные вопросы.

4 Составить отчет по лабораторной работе, содержащий:

- тему и цель работы;
- ход выполнения работы с приведением экранной формы работы программы;
- вывод по проделанной работе в соответствии с поставленной целью.

3.4 Варианты заданий

Варианты заданий для выполнения лабораторной работы представлены в таблице 3.1.

Таблица 3.1 – Варианты заданий для выполнения лабораторной работы

№ варианта	Диапазоны изменения случайных чисел					
	Зона S1			Зона S3		
	X	Y	Z	X	Y	Z
1	0.63 – 0.91	0.25 - 0.72	0.12 - 0.57	0.29 - 0.64	0.35 - 0.58	0.20 - 0.94
2	0.48 – 0.93	0.16 - 0.81	0.66 - 0.98	0.14 – 0.88	0.50 – 0.93	0.61 – 0.99
3	0.38 – 0.73	0.15 - 0.62	0.23 - 0.96	0.34 - 0.88	0.11 - 0.67	0.12 - 0.75
4	0.44 – 0.87	0.53 - 0.96	0.31 - 0.69	0.22 – 0.87	0.45 – 0.98	0.14 – 0.86
5	0.61 – 0.93	0.28 - 0.76	0.15 - 0.58	0.24 - 0.74	0.15 - 0.59	0.23 - 0.95
6	0.43 – 0.93	0.66 - 0.98	0.14 – 0.88	0.13 – 0.83	0.66 - 0.98	0.14 – 0.88
7	0.11 - 0.67	0.12 - 0.75	0.15 - 0.57	0.27 - 0.61	0.23 - 0.96	0.34 - 0.88
8	0.23 - 0.96	0.28 - 0.76	0.15 - 0.58	0.19 – 0.81	0.53 – 0.93	0.60 – 0.98
9	0.53 - 0.96	0.31 - 0.69	0.22 – 0.87	0.25 - 0.69	0.38 - 0.98	0.21 - 0.74
10	0.29 - 0.64	0.35 - 0.58	0.20 - 0.94	0.53 - 0.96	0.31 - 0.69	0.22 – 0.87

3.5 Контрольные вопросы

- 1 Для чего используется модель многокомпонентных объектов COM?
- 2 Какие технологии обеспечивают организацию взаимодействия приложений при управлении данными табличного процессора программой на языке высокого уровня?
- 3 Назовите достоинства и недостатки использования технологии OLE?
- 4 Назовите основные отличия реализации программы моделирования поведения мобильного объекта с помощью среды Lazarus от реализации в среде Excel с помощью макросов.
- 5 Применима ли технология OLE для управления приложениями в ОС Linux?

4 Лабораторная работа № 4 (4 часа). Идентификация положения мобильного объекта с использованием навигационной аппаратуры и табличного процессора

Цель работы: получить навыки по использованию табличного процессора для обработки навигационных данных.

4.1 Теоретическая часть

Рассмотренная модель идентификации положения мобильного объекта с учетом погрешностей в работах №1-№3 используют моделируемые различным способом входные данные. В данной работе приводится решение, позволяющее обрабатывать данные от навигационной аппаратуры с помощью табличного процессора. На рисунке 4.1 представлена структурная схема системы идентификации положения мобильного объекта с использованием навигационной аппаратуры.

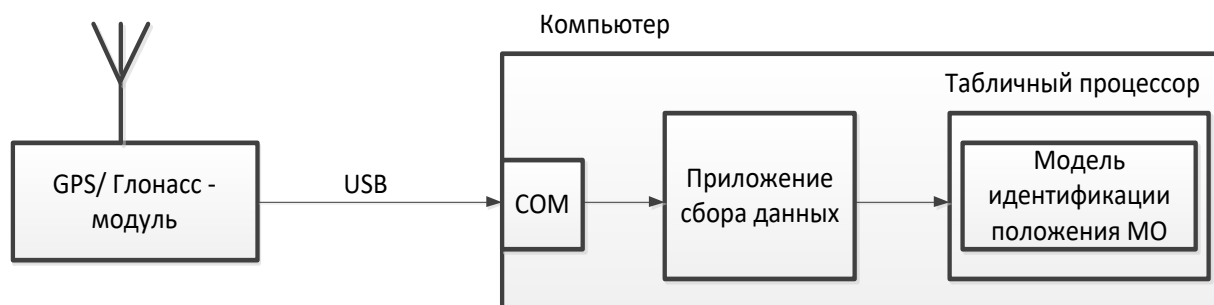


Рисунок 4.1 – Структурная схема системы идентификации положения МО

Входные данные для данной системы поступают от GPS/Глонасс- модуля, который рассчитывает на основе спутниковых сигналов положение мобильного объекта. Погрешность навигационного оборудования бюджетного ценового диапазона варьирует от 2.5 до нескольких десятков метров в зависимости от качества сигнала от спутниковой системы.

Для обмена данными с другими устройствами большинство навигационных средств используют UART-интерфейс. Типичной схемой подключения подобных устройств к компьютеру является следующая цепочка преобразований: UART-USB-

виртуальный COM-порт. Приложение по сбору данных получает навигационные данные от последовательного порта в формате NMEA, извлекает координаты и передает табличному процессору. На рисунке 4.2 представлено главное окно программы.

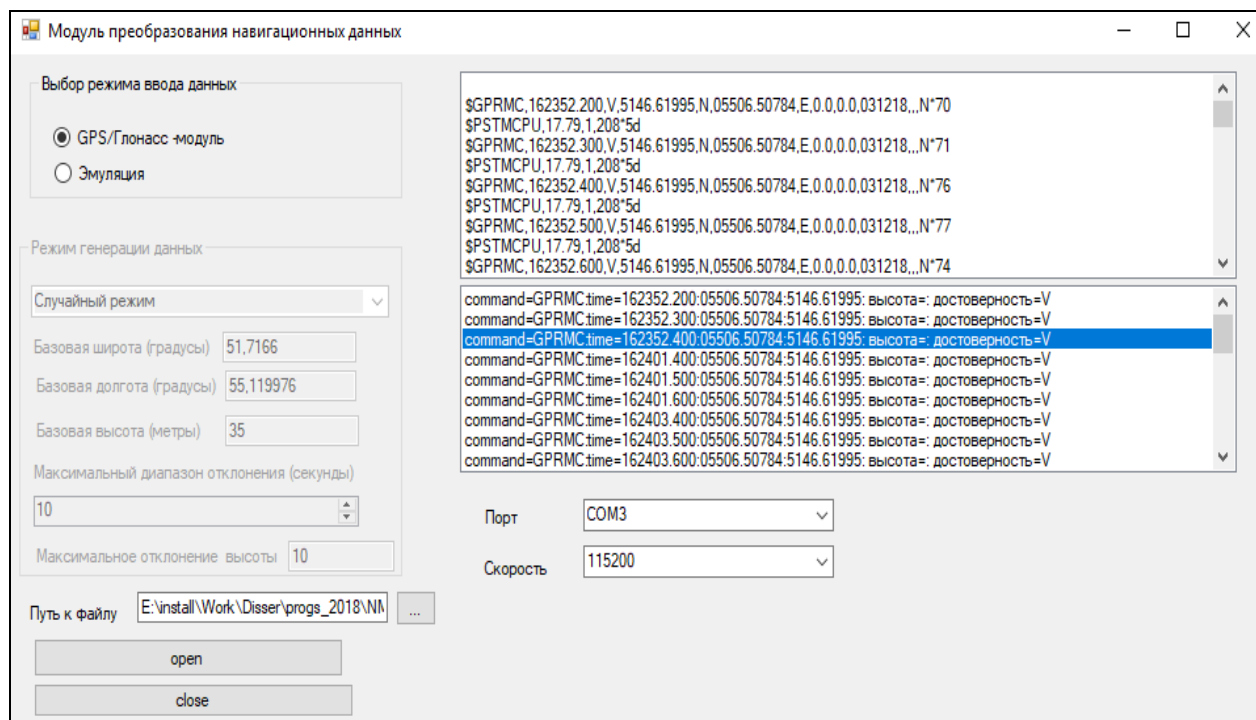


Рисунок 4.2 – Главное окно приложения сбора данных

Приложение позволяет получать данные как непосредственно от навигационного оборудования, так и эмулировать его работу (при отсутствии устройства). При работе с реальным оборудованием необходимо указать номер порта и скорость обмена. В поле «Путь к файлу» указывается расположение документа Excel, с реализованной моделью идентификации положения мобильного объекта. По нажатию кнопки «Open» начинается процесс преобразования. В окне отображаются данные в формате NMEA[6Д] и декодированные значения. В данном примере приложение тестировалось с использованием модуля SIM68[3Д].

На рисунке 4.3 представлен документ Excel, реализующий модель идентификации положения мобильного объекта.

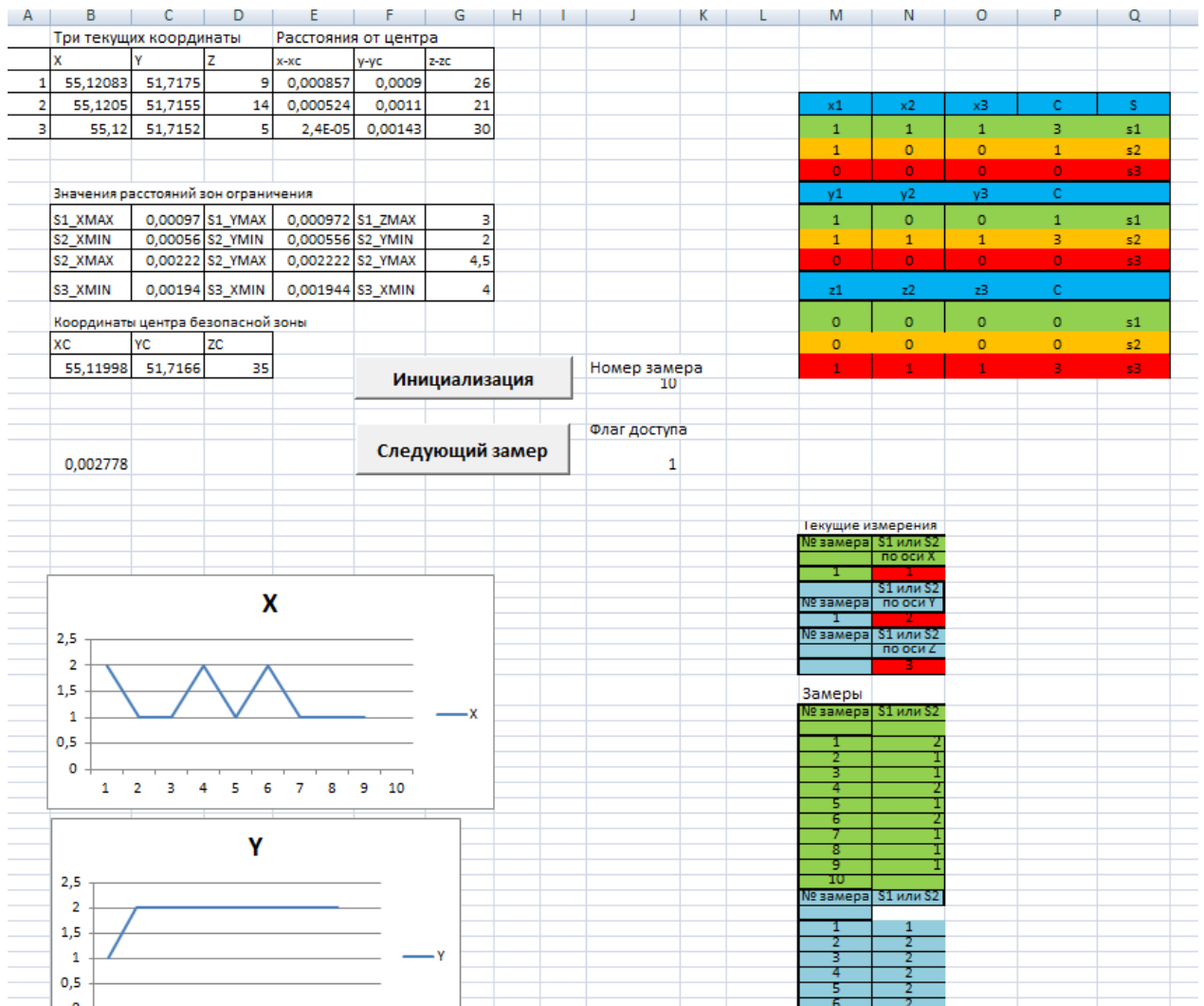


Рисунок 4.3 – Модель идентификации положения МО

Модель идентифицирует расположение МО в различных зонах относительно центральной точки. При определении зоны расположения МО учитываются текущие отклонения от центральной точки и ограничения зон. Для идентификации положения необходимо получение 3 координат. Ячейка «Флаг доступа» используется в служебных целях: для синхронизации табличного процессора и приложения сбора данных.

4.2 Постановка задачи

1 Изучить теорию об организации взаимодействия навигационного оборудования и табличного процессора.

2 Осуществить идентификацию положения мобильного объекта при помощи навигационного оборудования при различных размерах зон.

3 По результатам работы сделать выводы и составить отчет.

4 Ответить на контрольные вопросы.

4.3 Порядок выполнения работы

1 Подключите навигационное оборудование к компьютеру. При отсутствии – используйте режим эмуляции;

2 Откройте документ Excel и измените ограничения зон. После этого сохраните документ и закройте его.

3 Запустите приложение сбора данных, настройте его и запустите процесс преобразования.

4 В открытом документе Excel на кнопку «Следующий замер» переходите к следующей итерации определения местоположения МО.

5 Сделайте выводы по работе и составьте отчет по работе.

4.4 Контрольные вопросы

1 Как подключается обычно GPS/Глонасс –модуль к компьютеру?

2 Какое назначение программы сбора и преобразования навигационных данных?

3 Назначение протокола NMEA

4 Назовите типичные оценки погрешности навигационного оборудования.

5 Лабораторная работа № 5 (4 часа). Дистанционное изучение беспилотного мобильного объекта

Цель работы: получить представление о реализации системы дистанционного мониторинга за поведением беспилотного мобильного объекта с использованием канала удаленного видеонаблюдения.

5.1 Теоретическая часть

Под мобильным объектом информатизации (МОИ) в работе понимаются средства и системы обработки информации, используемые в соответствии с заданной информационной технологией, оснащенные системой глобального позиционирования, которые при эксплуатации меняют координаты своего местоположения. Одним из характерных представителей МОИ в настоящее время является класс беспилотных мобильных объектов (БМО), называемых дронами. Популярность и доступность дронов стремительно растёт. Это обусловлено широкими возможностями дронов применительно к различным областям деятельности и досуга человека.

Применение дронов обусловило целый ряд задач, среди которых наиболее актуальными являются задачи защиты от дронов и задачи защиты дронов. В данной работе приводится пример организации системы мониторинга за поведением БМО с использованием канала удаленного видеонаблюдения с целью удаленного изучения БМО, как объекта защиты [3].

В качестве БМО использовался квадрокоптер типа Parrot Bebop 2.

Технические характеристики квадрокоптера представлены в таблице 5.1.

Таблица 5.1 – Технические характеристики квадрокоптера Parrot Bebop Drone 2

Двигатель и шасси		Аккумулятор	
Тип двигателя	электрический	Емкость аккумулятора	2.7 Ач
Кол-во винтов	4 шт	Напряжение питания	11.1 В
Максимальная скорость	65 км/час	Макс. время полета	25 мин

Продолжение таблицы 5.1

Датчики		Модель аккумулятора	3S
Датчики	– высоты – GPS-модуль – гироскоп	Аккумуляторов	1 шт
Камера		Управление и передатчик	
Тип камеры	встроенная	Управление смартфоном	iOS/Android
Съемка Full HD (1080p)	1920x1080 пикс 30 к/с	Радиус действия	300 м
Кол-во мегапикселей	14	Частота радиоканала	2.4 GHz
Углы обзора	180°	Питание пульта ДУ	аккумулятор
Разрешение фото	4096x3072 пикс	Режимы полета	– функция возврата "домой", – акробатический режим

В основу работы положен принцип дистанционного мониторинга за поведением БМО с использованием удаленного видеонаблюдения. Актуальность данной работы определена необходимостью предоставления полной информации в учебную аудиторию в условиях дистанционного изучения объекта.

Структурная схема экспериментальной сети дистанционного видеонаблюдения за поведением БМО представлена на рисунке 5.1.

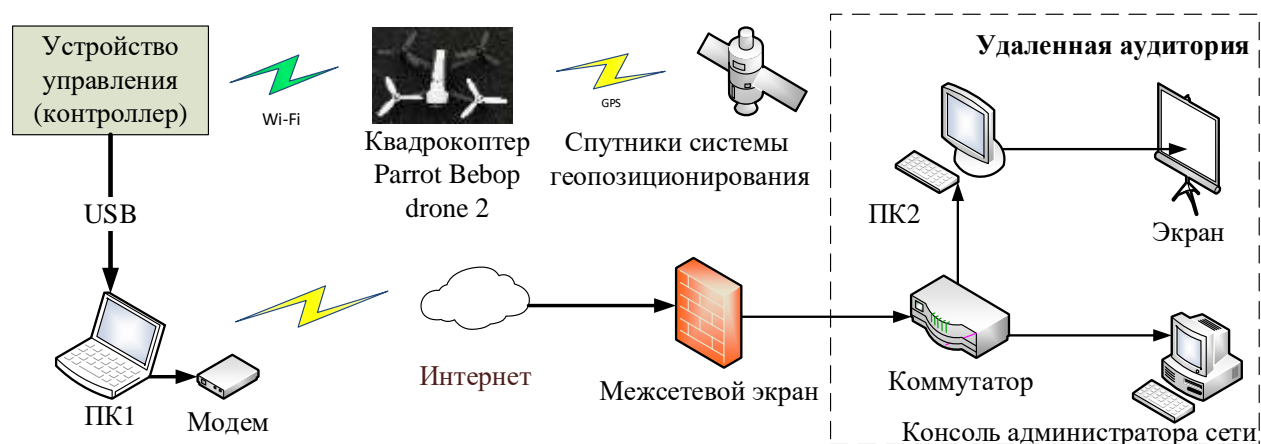


Рисунок 5.1 – Структурная схема системы удаленного видеонаблюдения за мобильным беспилотным объектом

Данная система позволяет транслировать видеосигнал квадрокоптера из специализированной лаборатории кафедры в одну из удаленных лекционных аудиторий, находящуюся в другом корпусе университета. Изображение, получаемое камерой квадрокоптера, передается с пульта управления квадрокоптером, находящегося в лаборатории кафедры, на настенный экран лекционной аудитории по сети Интернет. Передача информации на экран в лекционной аудитории осуществляется при помощи двух персональных компьютеров (ПК1 и ПК2 на рисунке 5.2) с предустановленной программой, обеспечивающей голосовую и видеосвязь.

На рисунке 5.2 представлено экранное видеоизображение объекта, передаваемое квадрокоптером в лекционную аудиторию в режиме реального времени.



Рисунок 5.2 – Экранное видеоизображение от объекта

Этот режим позволяет в условиях учебного процесса дистанционно регистрировать видеоданные от квадрокоптера, визуально наблюдать за его поведением в различных режимах и удаленно фиксировать основные

характеристики полета: скорость, высоту, уровень сигнала wi-fi и GPS, уровень заряда батареи, положение относительно объектов в помещении, как представлено на рисунке 5.3.



Рисунок 5.3 – Характеристики полета квадрокоптера

Достоинством предложенной системы является возможность дистанционного мониторинга за поведением БМО, удаленного наблюдения за его поведением в различных режимах, фиксирование основных характеристик полета и регистрации видеоданных от квадрокоптера.

5.2 Порядок выполнения работы

1. Подготовка к проведению лабораторной работы.

Для реализации системы удаленного видеонаблюдения за мобильным беспилотным объектом используются следующие технические средства:

- ноутбук с Wi-Fi адаптером с предустановленной ОС Windows для приема видеоизображения с пульта управления квадрокоптером (ПК 1);
- персональный компьютер для удаленной трансляции видеоизображения (ПК 2);
- смартфон с Android OS с предустановленной программой управления дроном;
- квадрокоптер Parrot Bebop 2.

Используемое программное обеспечение включает в себя:

- Freeflight Pro - мобильное приложение для управления дроном;
- MyPhoneExplorer - программа для синхронизации данных между смартфоном и ПК через USB интерфейс;

– Skype - программное обеспечение для удаленной видеотрансляции через Интернет.

В целях соблюдения техники безопасности необходим контроль отсутствия людей, не связанных с проведением эксперимента, в помещении, где осуществляется перемещение МО.

2. Порядок операций

2.1 Для выполнения лабораторной работы необходимо предварительно скачать и установить предложенные программные средства:

– Freeflight Pro - устанавливается на устройство, используемое в качестве пульта управления (ПУ) квадрокоптером (смартфон, планшет и т.д.); приложение бесплатно и доступно в магазинах приложений Play market и App Store;

– MyPhoneExplorer клиента - устанавливается на пульт управления и MyPhoneExplorer - устанавливается на ПК, ПО доступно на официальном сайте разработчика по ссылке: <https://www.fjsoft.at/en/downloads.php> (для ПК) и в магазине приложений Play market (для пульта управления);

– Skype - ПО доступно на официальном сайте разработчика по ссылке: <https://www.skype.com/ru/get-skype/>.

Далее необходимо настроить трансляцию изображения с экрана пульта управления на экран компьютера, передающего видеоизображение.

На ПУ необходимо произвести следующие настройки (далее приводится информация для настройки устройств с операционной системой Android):

– убедиться, что установлены драйверы интерфейса отладки (ADB-драйверы, обычно устанавливаются при первом подключении устройства к ноутбуку, так же доступны в сети Интернет отдельно под конкретную модель).

– убедиться, что на устройстве включена USB-отладка ("Настройки > Приложения > Разработка > Отладка USB" для Android 2+ или "Настройки > Для разработчиков > Отладка по USB" для Android 4+). В Android 4.2.2 и более новых версиях нужно предварительно разблокировать опцию USB-отладки (Настройки > О телефоне > Номер сборки > коснуться 7 раз). Далее в разделе специальных возможностей необходимо выбрать пункт для разработчиков (перейти в Настройки

– Для разработчиков) и разрешить «Отладку по USB», поставив галочку напротив соответствующего пункта.

2.2.2 Физически соединить пульт управления с ноутбуком при помощи USB-кабеля, после чего на ПУ в появившемся окне настройки соединения выбрать пункт «Только зарядка».

2.2.3 Для синхронизации данных между ПК и ПУ необходимо запустить приложение "MyPhoneExplorer клиент" на телефоне "MyPhoneExplorer» на ПК и выбрать тип соединения по USB. Для выбора типа соединения в приложении на ПК нужно перейти "Файл > Настройки" и в группе "Соединение через..." установить переключатель в положение "USB-кабель".

2.2.4 В программе, установленной на ПК добавить ПУ в базу данных программы, нажав «Пользователь-Добавить пользователя» и выбрать тип соединения через USB. После произведенных действий необходимо дождаться сопряжения устройств.

2.2.5 Для подключения к ПУ в главном окне "MyPhoneExplorer" необходимо нажать "Файл > Подключение". После того, как связь будет установлена, синхронизировать содержимое устройства с компьютером. Для вывода экрана телефона на экран компьютера перейти "Разное > Клавиатура телефона/Зеркалирование экрана", как показано на рисунке 5.4.

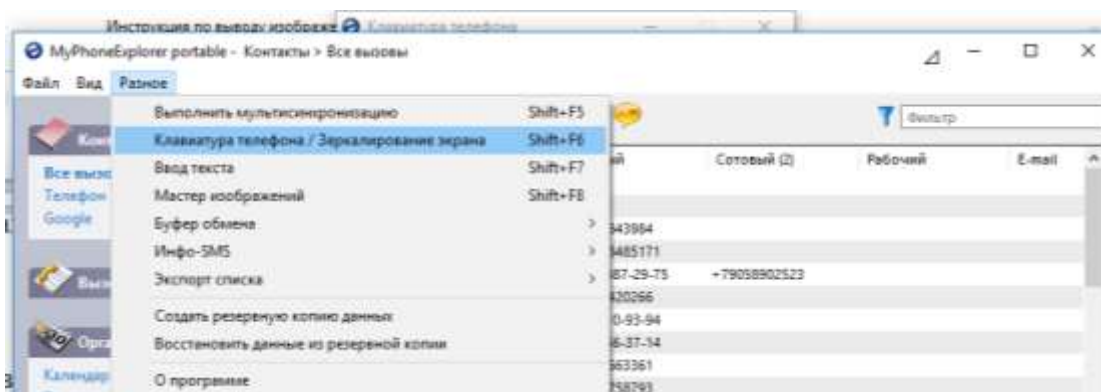


Рисунок 5.4 – Синхронизация данных между ПК и пультом управления квадрокоптера

2.3 Для вывода видеоизображения квадрокоптера на экран ПК необходимо запустить на телефоне приложение для управления квадрокоптером Freeflight Pro и подключиться к квадрокоптеру по сети wi-fi (сеть Вебор Drone 2), после чего изображение с камеры квадрокоптера будет транслироваться на экран ноутбука, как показано на рисунке 5.5.



Рисунок 5.5 – Трансляция видеоизображения с квадрокоптера на экран ПК

2.4 Передача видео потока от ПК1 к ПК2 осуществляется с помощью ПО Skype, устанавливаемого на компьютеры, между которыми необходимо настроить видеосвязь. Для установления видеосвязи необходимо авторизоваться в аккаунте Skype, либо зарегистрироваться в случае его отсутствия. После входа в аккаунт – установить соединение, нажав на кнопку видеозвонка, как показано на рисунке 5.6.

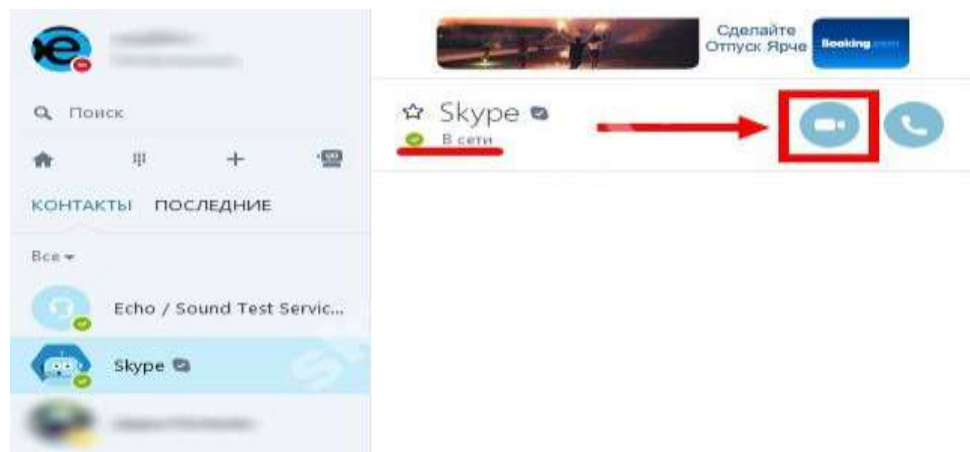


Рисунок 5.6 - Установка видео соединения между ПК 1 и ПК2

Для вывода изображения с экрана рабочего стола ПК 1 на экран ПК 2 необходимо перейти в пункт меню «Звонки – Демонстрация экрана».

В результате выполненных действий изображение с квадрокоптера будет транслироваться на экран удаленного ПК.

5.3 Задание на лабораторную работу

1 Ознакомиться с теоретическим материалом по лабораторной работе.

2 Построить систему дистанционного видеонаблюдения за поведением беспилотного мобильного объекта, используя оборудование и программное обеспечение, предложенное преподавателем и следуя инструкциям пункта 5.2.

3 Проанализировать полученные характеристики полета квадрокоптера (скорость, высоту, уровень сигнала wi-fi и GPS, уровень заряда батареи, положение относительно объектов в помещении). Сделать вывод о достоинствах и недостатках построенной системы.

4 Устно ответить на контрольные вопросы.

5 Составить отчет по лабораторной работе, в котором указать:

- тему и цель работы;
- ход работы;
- вывод по проделанной работе в соответствии с поставленной целью.

5.4 Контрольные вопросы

1 Что такое мобильный объект информатизации?

2 Какие задачи решает система дистанционного мониторинга за поведением беспилотного мобильного объекта?

3 Назовите основное оборудование и программное обеспечение, необходимое для построения системы дистанционного мониторинга за поведением БМО.

4 Какие характеристики полета квадрокоптера можно получить при использовании системы удаленного видеомониторинга?

6 Лабораторная работа № 6 (4 часа). Исследование сетевого трафика беспилотного мобильного объекта

Цель работы: получить представление о реализации дистанционной системы управления беспилотным мобильным объектом.

6.1. Теоретическая часть

Широкое применение дронов в различных областях деятельности человека определяет актуальность обеспечения их информационной безопасности. Одной из актуальных угроз является отправка несанкционированных команд со стороны злоумышленника. Для противодействия подобным угрозам требуется своевременное обнаружение атак. Одним из подходов решения данной задачи является анализ сетевого трафика, направленного на беспилотный мобильный объект БМО.

В данной работе исследуется возможность использования данных сетевого трафика для анализа поведения БМО с целью выявления аномальных ситуаций. На рисунке 6.1 представлена схема лабораторного комплекса для сбора сетевого трафика дрона Parrot Bebop 2.

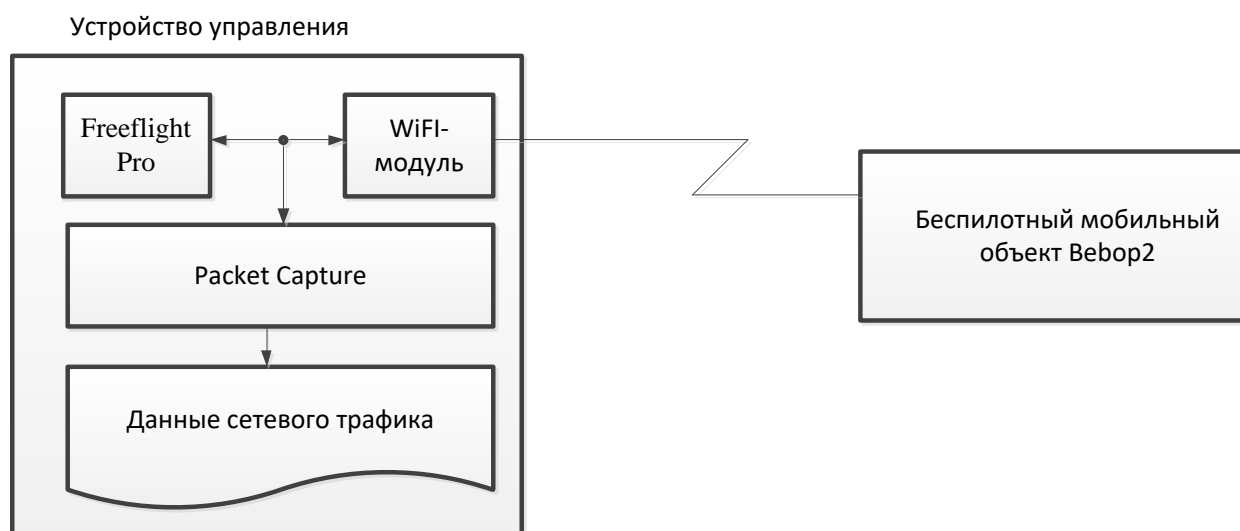


Рисунок 6.1 – Структурная схема лабораторного комплекса

Устройство управления подключается к точке доступа дрона по WiFi и передает команды с помощью программы FreeFlight Pro. Одновременно сетевой трафик перехватывается утилитой Packet Capture[9Д] и сохраняется в файл. После проведенного эксперимента сохраненные данные анализируются для выявления поведения мобильного объекта.

На рисунке 6.2 представлены результаты перехвата сетевого трафика, полученные при следующих условиях: - беспилотный мобильный объект имеет IP-адрес 192.168.42.1; - смартфон, используемый как пульт управления, имеет IP-адрес 192.168.42.33.

The screenshot shows a Wireshark interface with a packet list table and a packet details pane. The packet list table contains the following data:

No.	Time	Source	Destination	Protocol	Length	Info
7625	16.315970	192.168.42.1	192.168.42.33	UDP	939	5004 → 55004 Len=895
7626	16.316724	192.168.42.1	192.168.42.33	UDP	849	5004 → 55004 Len=805
7627	16.316833	192.168.42.1	192.168.42.33	UDP	1014	5004 → 55004 Len=970
7628	16.316944	192.168.42.1	192.168.42.33	UDP	855	5004 → 55004 Len=811
7629	16.316995	192.168.42.1	192.168.42.33	UDP	549	5004 → 55004 Len=505
7630	16.323726	192.168.42.33	192.168.42.1	RTCP	104	Receiver Report Sour
7631	16.330611	192.168.42.33	192.168.42.1	UDP	64	43341 → 54321 Len=20
7632	16.347070	192.168.42.1	192.168.42.33	UDP	82	47075 → 54321 Len=38
7633	16.347193	192.168.42.1	192.168.42.33	UDP	845	5004 → 55004 Len=801
7634	16.347737	192.168.42.1	192.168.42.33	UDP	990	5004 → 55004 Len=946
7635	16.348488	192.168.42.1	192.168.42.33	UDP	1011	5004 → 55004 Len=967
7636	16.348608	192.168.42.1	192.168.42.33	UDP	939	5004 → 55004 Len=895
7637	16.348800	192.168.42.1	192.168.42.33	UDP	939	5004 → 55004 Len=895
7638	16.348863	192.168.42.1	192.168.42.33	UDP	1002	5004 → 55004 Len=958
7639	16.348901	192.168.42.1	192.168.42.33	UDP	814	5004 → 55004 Len=770
7640	16.348938	192.168.42.1	192.168.42.33	UDP	617	5004 → 55004 Len=573
7641	16.349333	192.168.42.1	192.168.42.33	UDP	818	5004 → 55004 Len=774
7642	16.349472	192.168.42.1	192.168.42.33	UDP	817	5004 → 55004 Len=773
7643	16.349972	192.168.42.1	192.168.42.33	UDP	831	5004 → 55004 Len=787

The packet details pane for frame 7632 shows:

- Frame 7632: 82 bytes on wire (656 bits), 82 bytes captured (656 bits)
- Linux cooked capture
- Internet Protocol Version 4, Src: 192.168.42.1, Dst: 192.168.42.33
- User Datagram Protocol, Src Port: 47075, Dst Port: 54321
- Data (38 bytes)

The raw data section shows the following hex and ASCII representation:

```

0000  00 00 00 01 00 06 a0 14 3d c3 08 dc 5f a4 08 00  ..... =...
0010  45 c0 00 42 28 be 40 00 40 11 3b ba c0 a8 2a 01  E..B( .@. @.;...*
0020  c0 a8 2a 21 b7 e3 d4 31 00 2e 7d b3 02 7f f7 26  ..*!...1 .}....&
0030  00 00 00 01 04 09 00 e0 16 c4 2d 0c ea 49 40 07  ..... I@.
0040  5f 51 37 3d 96 4b 40 00 00 00 60 66 a6 61 40 00  _Q7= .K@. ...f.a@.
0050  00 02  ..
  
```

Рисунок 6.2 – Данные сетевого трафика БМО

В зависимости от этапа взаимодействия с устройством Webor2 выделяются три типа используемых протоколов:

– на этапе поиска и обнаружения БМО применяется протокол mDNS, который позволяет определить IP-адрес и сервисный порт квадрокоптера;

– этап соединения. В данном режиме используется протокол TCP с ранее определенным сервисным портом. По каналу передается информация в виде строки формата JSON, использующая для определения параметров протокола UDP в основном режиме работы;

– основной режим работы. На данном этапе БПМО Webor2 использует протокол UDP для передачи команд управления и данных, в том числе видеоданных. Описание протокола на прикладном уровне представлено в документе ARSDK Protocols [7Д].

В рамках данной работы основное внимание уделено протоколу третьего класса. Протокол прикладного уровня квадрокоптера состоит из двух частей: заголовка и непосредственно данных. Заголовок определяет функциональное назначение данного кадра и состоит из 4 полей:

- типа данных (1 байт);
- идентификатор буфера назначения (1 байт);
- порядковый номер обращения к буферу;
- размер всего кадра.

При этом поддерживается 4 типа данных:

- ACK (1) для подтверждения ранее полученных данных;
- Data (2) для передачи обычных данных без подтверждения;
- Low latency data (3) для передачи данных с высоки приоритетом;
- Data with Ack (4) для передачи данных с требованием подтверждения. На рисунке 3 видно, что первый байт в области данных UDP-пакета равен 02h и определяет передачу обычных данных.

Фактически при передаче данных по сети протокол осуществляет запись в буфер, который можно рассматривать как память типа FIFO, копия которой находится у получателя пакета. При этом каждый класс имеет различные характеристики: по размеру буфера и временным параметрам. С каждым буфером связан счетчик, который увеличивается при каждой передаче данных и используется для выявления повторных пакетов.

В области непосредственных данных кодируются команды или данные. Тип передаваемой информации определяется типом используемого буфера. Например, буфер с идентификатором 127 используется для передачи данных класса ARCommands от дрона к устройству управления.

Команда класса ARCommands, в свою очередь, имеет следующую структуру:

- идентификатор проекта (1 байт);
- идентификатор класса команд (1 байт);
- идентификатор непосредственно команды (2 байта).

Необходимо учитывать, что стек протоколов ARSDK используется для различных типов устройств. Поэтому идентификатор проекта используется для определения набора команд конкретного устройства. Для БМО Вебор2 идентификатор проекта равен 1 и имеет имя «adrone3».

Все команды разделены на группы по функциональному назначению. Например, класс команд «Pilotingstate» с идентификатором «4» содержит следующие команды:

- Flat trim changed (id=0);
- Flying state(id=1);
- Alert state (id=2);
- GpsLocationChanged (id=9).

Потом идут аргументы команды. Каждая команда характеризуется индивидуальным набором аргументов, которые отличаются как по количеству, так и по типу закодированной информации.

Ниже приведены аргументы команды GpsLocationChanged (изменение навигационных координат):

- широта (double), градусы;
- долгота (double), градусы;
- высота (double), метры;
- погрешность определения широты (uint8);
- погрешность определения долготы (uint8);
- погрешность определения высоты (uint8).

Подробную информацию о командах ARCommands для устройства Bebop2 можно узнать из документа `adrone3.xml`[8Д].

Пример декодирования UDP-пакета под номером 7632 (рисунок 6.2) представлены в таблице 6.1.

Таблица 6.1 – Результаты декодирования команды

Значение, hex	Значение, dec	Назначение	Текущее значение
Заголовок прикладного протокола квадрокоптера			
02	2	тип данных	передаются данные без подтверждения
7F	127	тип буфера	передаются данные в формате ARCommands от дрона к пульту управления
F7	247	номер последовательности	идентификатор кадра, переданного по данному буферу
26 00 00 00	38	размер кадра (данные представлены в обратном направлении)	38 байт
ARCommands			
01	1	Идентификатор проекта	Adrone3
04	4	Идентификатор группы команд	Pilotingstate
09 00	9	Идентификатор команды	PositionChanged
Аргументы			
E0-16-C4-2D-0C-EA-49-40	51,828497	Широта(тип double)	51,82849666666668
07-5F-51-37-3D-96-4B-40	55,173743	Долгота (тип double)	55,173743166668
00-00-00-60-66-A6-61-40	141,199997	Высота (тип double)	141,199996948242
00	0	погрешность определения широты (м.)	0
00	0	погрешность определения долготы (м.)	0
02	2	Погрешность определения высоты (м.)	2

Для автоматизации декодирования и изучения сетевых пакетов разработана программа `DecodeDronePackets`, которая обрабатывает файл с перехваченным сетевым трафиком.

На рисунке 6.3 представлена экранная форма программы и сохраненные результаты в текстовый файл.

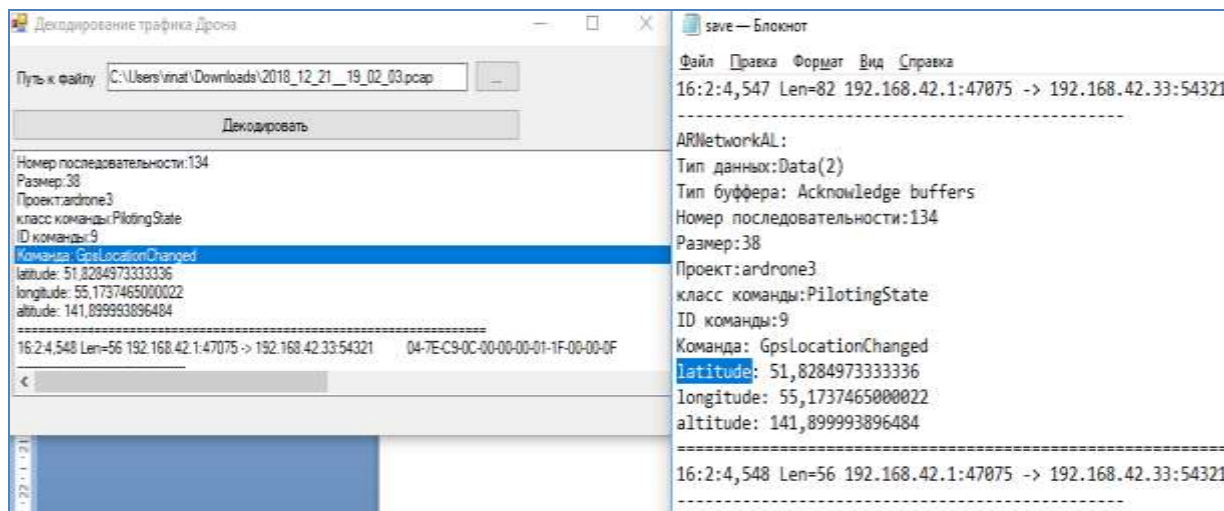


Рисунок 6.3 – Экранная форма программы декодирования пакетов сетевого трафика дрона

Результаты данной методики позволяют изучить сетевые протоколы обмена информацией с БМО, наблюдать в режиме реального времени за маршрутом его движения по навигационным данным, а также могут использоваться для обнаружения сетевых атак на подсистему контроля и управления объектом.

6.2 Постановка задачи

1 Изучить теорию о протоколах, используемых для взаимодействия дрона и устройства управления.

2 Получить файл с данными сетевого трафика дрона в формате rcсар. И декодировать данные в формате ARCommands от дрона к пульта управления. Результаты представить в таблицу 6.1

3 При помощи программы DecodeDrone декодировать данные сетевого трафика. На основе 10 последовательных команд сделать выводы о поведении дрона.

4 По результатам работы сделать выводы и составить отчет.

5 Ответить на контрольные вопросы.

6.3 Порядок выполнения работы

1 Открыть файл в программе WireShark. Найдите пакет, в которой передаются данные в формате ARCommands от дрона к пульта управления. Данные пакеты обладают следующими характеристиками:

- UDP-протокол;
- IP-адрес источника -192.168.42.1;
- размер пакета меньше 100 байт (передаются команды, а не видеокадры).

2 Выделите выбранный UDP-пакет и его область данных в средней части окна, как показано на рисунке 6.1. Убедитесь, что первые два байта в области данных UDP-пакета имеют значения 02 и 7F (передача данных без подтверждения от дрона к пульта в формате ARCommands). Иначе, продолжите поиск пакетов соответствующего типа.

3 Декодируйте пакет и представьте результат в виде таблицы, аналогичной таблице 6.1.

4 Опишите, назначение передаваемых данных.

5 Запустите программу DecodeDrone и декодируйте файл с сетевым трафиком.

6 Выберите в произвольной временной области 10 последовательных команд от пульта управления к дрону и заполните представленную ниже таблицу 6.2.

Таблица 6.2 – Команды управления

№	Направление: от пульта к дрону	Класс команды	Идентификатор команды	Описание
1	2	3	4	5
1	Да	Piloting	2	Move the drone
2	Да	Piloting	2	Move the drone
3	Нет	PilotingState	5	Drone's speed changed
4	Да	Piloting	2	Move the drone
5	Нет	PilotingState	6	Drone's attitude changed
6	Да	Piloting	2	Move the drone
7	Нет	CameraState	0	Camera orientation

Продолжение таблицы 6.2

1	2	3	4	5
8	Да	Piloting	2	Move the drone
9	Да	Piloting	2	Move the drone
10	Да	Piloting	2	Move the drone

Сделайте вывод об изменении в поведении дрона.

7 Сделайте выводы по работе и составьте отчет по работе.

6.4 Контрольные вопросы

- 1 Через какой канал связи подключаются пульт управления к дрону Bebop2.
- 2 Какие два основных класса протоколов прикладного уровня используются для взаимодействия с дроном.
- 3 Какой тип сетевого протокола используется на этапе подключения.
- 4 Какие поля содержит заголовок протокола ARSDK Protocols.
- 5 Приведите примеры команд ARCommands для дрона Bebop2.

7 Лабораторная работа № 7 (4 часа). Метод ранжирования рисков на основе ассоциативного подхода

Цель работы: освоить процедуру автоматизированного ранжирования рисков от угроз на основе ассоциативного подхода.

7.1 Теоретическая часть

В основу модели ранжирования рисков положен принцип ассоциативности значения величины риска $\langle zR_i \rangle$ от угрозы, имеющей индекс (номер) i - с порядковым номером ранга этой величины J в ранжированном ряду рисков S [1,2].

В терминах теории распознавания образов процедура определения номера риска Q^* в ранжированном ряду S для исследуемого значения риска q^x описывается следующим образом [1Д]:

$$Q = \{Q_1, Q_2, \dots, Q_N\}; D(\langle zR_i \rangle) = \{d_1, d_2, \dots, d_N\}; P = \{\langle zR_i \rangle\}; \quad (1)$$

$$KP: q^x \in Q^* \in Q: \langle zR_i \rangle \in dQ^*, \quad (2)$$

где Q , P , D и KP – соответственно: множество номеров рангов в ряду S , множество признаков распознавания, множество поддиапазонов значений рисков, соответствующих порядковому номеру J , и классифицирующее правило.

В соответствии с выражением (2) исследуемый образ q^x (или же некоторое значение риска $\langle zR_i \rangle$) относится к классу Q^* (или рангу Q^*) из множества рангов Q , если это значение риска $\langle zR_i \rangle$ входит в диапазон значений d рисков, ранжируемых под номером Q^* .

В процессе идентификации параметра J учитывается, что нескольким значениям рисков, принадлежащих одному диапазону, первоначально присваивается один и тот же номер ранга в ряду J . В этом случае при определении конечного номера ранга J учитывается номер по порядку поступления значения риска. Применение ассоциативного принципа позволяет определять ранг риска за один такт времени, синхронный с его поступлением.

Программа «Ранжирование рисков от угроз на основе ассоциативного принципа» предназначена для имитационного моделирования и исследования метода ранжирования рисков в задачах исследования модели угроз информационной безопасности. Программа также может найти применение в системах реального времени, где требуется высокая оперативность построения ранжированного ряда. Высокая производительность работы программы обеспечивается применением ассоциативного принципа ранжирования, позволяющего, в отличие от известных алгоритмов, при поступлении данных по риску одновременно определять ранг этих данных в ранжированном ряду. Основной функцией программы «Ранжирование рисков от угроз на основе ассоциативного принципа» является сортировка рисков по возрастанию.

Ранжирование рисков производится в двух режимах:

- в режиме автоматического ввода с использованием генератора случайных чисел (режим **A**);
- в режиме ручного ввода значений рисков (режим **B**).

Входными данными алгоритма являются количество обрабатываемых рисков или количество угроз NU , максимальное значение риска $\langle zR_{max} \rangle$ и, непосредственно, значения оценок рисков $\langle zR_i \rangle$. Значения рисков обрабатываются последовательно в порядке их поступления и на основе ассоциативного принципа их номера размещаются в матрице, имитирующей ассоциативную память. В начале работы программы определяется максимальное значение величины вводимого риска $\langle zR_{max} \rangle$ и величины поддиапазонов значений рисков для каждого ранга ряда S . При этом различие между двумя соседними поддиапазонами (дискретность ряда диапазона) Δd определяется как отношение $\langle zR_{max} \rangle$ к N , а нижняя и верхняя границы поддиапазонов D_i для каждого ранга NR_i , соответственно, как $(NR-1)*\Delta d$ и $(NR-1)*\Delta d + \Delta d$.

В основу ранжирования положен принцип ассоциации каждого значения $\langle zR_i \rangle$ с соответствующей ячейкой ассоциативной памяти (АП), адрес которой NA соответствует номеру поддиапазона, в который входит значение $\langle zR_i \rangle$. Каждой ячейке ассоциативной памяти соответствует счетчик ассоциаций значений $\langle zR_i \rangle$

конкретному адресу АП. Ранжированный список величин рисков формируется по данным АП и счетчиков в порядке убывания величины риска. При этом самому большому значению риска и, соответственно, угрозе вызывающей этот риск, присваивается номер 1, а самому маленькому значению риска и его угрозе – номер *N*.

В конце работы алгоритма выводится ранжированный список индексов рисков угроз информации в исследуемой системе.

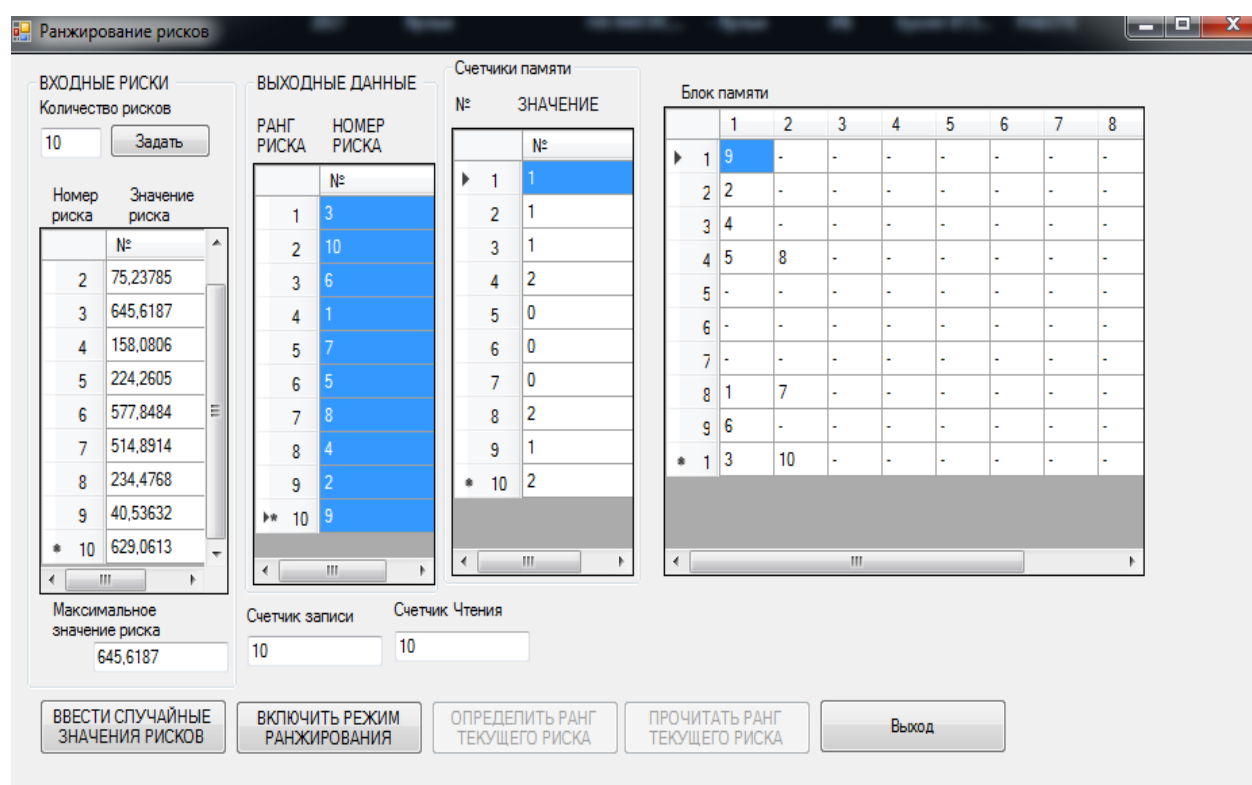


Рисунок 7.1 – Экранная форма программы ранжирования рисков и угроз в режиме А

Достоинством рассматриваемого метода ранжирования является высокая производительность, позволяющая в режиме реального времени определять ранжированный ряд рисков, управлять принятием решений в соответствии с приоритетами угроз.

7.2 Порядок выполнения работы

1.2.1 Режим А.

На рисунке 7.1 представлена экранная форма для демонстрации режима А. Выполнение ранжирования производится в следующем порядке:

- задается число N , например 10;
- вводятся случайные значения рисков и определяется автоматически $\langle zR_{max} \rangle$;
- включается режим ранжирования, при этом производится последовательная запись значений рисков в АП по соответствующим адресам;
- производится заполнение счетчиков по числу ассоциаций значений рисков с адресами АП;
- производится чтение данных проранжированного ряда.

1.2.2. Режим Б.

Ранжирование в ручном режиме по последовательности выполняемых операций аналогично режиму А. В процессе ранжирования необходимо правильно выбрать и занести $\langle zR_{max} \rangle$.

Завершение работы программы производится клавишей «Выход»

7.3 Задание на лабораторную работу

- 1 Ознакомиться с моделью ранжирования рисков угроз;
- 2 В соответствии с заданным вариантом составить алгоритм ранжирования рисков угроз;
- 3 С использованием программы провести ранжирование рисков угроз.
- 4 Просканировать экранную форму результата ранжирования, скопировать ее и перенести в отчет по работе.
- 5 Ответить на контрольные вопросы.

7.4 Варианты заданий

Варианты заданий для выполнения лабораторной работы представлены в таблице 7.1.

Таблица 7.1 – Варианты заданий для выполнения лабораторной работы

Вариант	Режим ранжирования	Число угроз	Максимальное значение риска
---------	--------------------	-------------	-----------------------------

1	A	8	1000
2	B	10	2000
3	A	12	500
4	B	10	1200
5	A	12	800
6	B	8	600
7	A	10	1500
8	B	11	1000
9	A	12	1400
10	B	15	2000

7.5 Контрольные вопросы

- 1 Дайте определение угрозы, риска, ущерба.
- 2 Приведите примеры задач с использованием изученного метода.
- 3 Смоделируйте режим, в котором ранжированный ряд содержал 10 рисков 2 рангов.
- 4 Смоделируйте режим, в котором ранжированный ряд содержал 10 рисков 10 рангов.
- 5 Можно ли смоделировать режим, в котором ранжированный ряд содержал 10 рангов одного ряда?

8 Лабораторная работа № 8. Метод оценки положения МО в трехмерном пространстве с использованием микроконтроллера ADuC812 и теории нечеткой логики

Цель работы: освоить алгоритм оценки положения МО с использованием теории нечеткой логики и его аппаратно-программную реализацию на основе микроконтроллера ADuC812

8.1 Теоретическая часть.

Для описания задачи идентификации зоны мобильного объекта (МО) при построении математической модели в терминах нечеткой логики использованы следующие условные обозначения [4Д]:

- $S = \{S_1, S_2, S_3\}$ – множество контролируемых зон МО, идентифицируемых преобразователем по нечеткой логике (ПНЧЛ);
- q^x – неизвестный образ зоны, подлежащий идентификации;
- S^* – образ зоны, к которому отнесен q^x из множества S ;
- $P = \{X, Y, Z\}$ – множество признаков идентификации образов, представляющий множество координат в пространстве;
- $\langle p_i \rangle$ – зарегистрированное значение i -го признака ($i = 1, 3$) по координатам x, y и z ;
- $\langle P^x \rangle$ – вектор зарегистрированных значений признаков q^x , например:
 $\langle P^x \rangle = \{\langle x \rangle, \langle y \rangle, \langle z \rangle\}$;
- $D = \{DX, DY, DZ\}$ – множество диапазонов изменения признаков, состоящее из множеств поддиапазонов признаков для каждого образа из множества S ;
- $DX = \{DXS_1, DXS_2, DXS_3\}$;
- $DY = \{DYS_1, DYS_2, DYS_3\}$;
- $DZ = \{DZS_1, DZS_2, DZS_3\}$.

Для идентификации состояния МО вследствие пересечения зон, обусловленного погрешностями навигаторов (см. рис. 1.1 и 1.2), используется двухступенчатая система продукционных правил.

На первой ступени определяется состояние q^x МО в соответствии с выражениями (10)-(12).

$$q^x(1,1) = S1, \text{ ЕСЛИ } (\langle x1 \rangle \in DXS1 \cap \langle x2 \rangle \in DXS1 \cup \langle x1 \rangle \in DS1 \cap \langle x3 \rangle \in DS1 \cup \langle x2 \rangle \in DS1 \cap \langle x3 \rangle \in DS1) \cup (\langle y1 \rangle \in DYS1 \cap \langle y2 \rangle \in DYS1 \cup \langle y1 \rangle \in DYS1 \cap \langle y3 \rangle \in DYS1 \cup \langle y2 \rangle \in DYS1 \cap \langle y3 \rangle \in DYS1) \cap (\langle z1 \rangle \in DZS1 \cap \langle z2 \rangle \in DZS1 \cup \langle z1 \rangle \in DZS1 \cap \langle z3 \rangle \in DZS1 \cup \langle z2 \rangle \in DZS1 \cap \langle z3 \rangle \in DZS1); \quad (10)$$

$$q^x(1,2) = S2, \text{ ЕСЛИ } (\langle x1 \rangle \in DXS2 \cap \langle x2 \rangle \in DXS2 \cup \langle x1 \rangle \in DS2 \cap \langle x3 \rangle \in DS2 \cup \langle x2 \rangle \in DS2 \cap \langle x3 \rangle \in DS2) \cup (\langle y1 \rangle \in DYS2 \cap \langle y2 \rangle \in DYS2 \cup \langle y1 \rangle \in DYS2 \cap \langle y3 \rangle \in DYS2 \cup \langle y2 \rangle \in DYS2 \cap \langle y3 \rangle \in DYS2) \cap (\langle z1 \rangle \in DZS2 \cap \langle z2 \rangle \in DZS2 \cup \langle z1 \rangle \in DZS2 \cap \langle z3 \rangle \in DZS2 \cup \langle z2 \rangle \in DZS2 \cap \langle z3 \rangle \in DZS2); \quad (11)$$

$$q^x(1,3) = S3, \text{ ЕСЛИ } (\langle x1 \rangle \in DXS3 \cap \langle x2 \rangle \in DXS3 \cup \langle x1 \rangle \in DS3 \cap \langle x3 \rangle \in DS3 \cup \langle x2 \rangle \in DS3 \cap \langle x3 \rangle \in DS3) \cup (\langle y1 \rangle \in DYS3 \cap \langle y2 \rangle \in DYS3 \cup \langle y1 \rangle \in DYS3 \cap \langle y3 \rangle \in DYS3 \cup \langle y2 \rangle \in DYS3 \cap \langle y3 \rangle \in DYS3) \cap (\langle z1 \rangle \in DZS3 \cap \langle z2 \rangle \in DZS3 \cup \langle z1 \rangle \in DZS3 \cap \langle z3 \rangle \in DZS3 \cup \langle z2 \rangle \in DZS3 \cap \langle z3 \rangle \in DZS3); \quad (12)$$

Вторая ступень продукционных правил используется для уточнения результатов идентификации состояния в случае возникновения двойственных решений на первой ступени, когда по навигационным данным МО находится одновременно в двух соседних зонах: $S1$ и $S2$, или $S2$ и $S3$. В данных случаях считаем, что МО находится в более опасной зоне, соответственно, в $S2$ или $S3$. Продукционные правило второй ступени имеют следующий вид:

$$\begin{aligned} q^x(2,1) &= S2, \text{ ЕСЛИ } q^x(1,1) = q^x(1,2); \\ q^x(2,2) &= S3, \text{ ЕСЛИ } q^x(1,2) = q^x(1,3). \end{aligned} \quad (13)$$

В процессе идентификации состояний МО сначала производится получение результатов по формулам (10) – (12) и далее при необходимости – по формуле (13).

Обобщенная схема алгоритма оценки положения МО на микроконтроллере представлена на рисунке 8.1.

Входными данными алгоритма являются три координаты мобильного объекта в трехмерном пространстве. Далее в цикле подсчитывается число попаданий координат в три зоны (S_1, S_2, S_3) по каждой оси системы координат и результаты сохраняются в массивы счетчиков:

- s_x – массив счетчиков трех зон по оси X ;
- s_y – массив счетчиков трех зон по оси Y ;
- s_z – массив счетчиков трех зон по оси Z .

На следующем шаге определяются оценки состояния по каждой оси по номеру счетчика с максимальным значением соответствующего блока счетчиков:

- S_x – номер состояния по оси X ;
- S_y – номер состояния по оси Y ;
- S_z – номер состояния по оси Z .

Работа проводится на стенде SDK1.1, основу которого представляет 8-битный микроконтроллер ADuC812. Данный микроконтроллер умеет выполнять базовые арифметические и логические команды, в том числе умножения и деления.

Объем доступной памяти для программ в рамках стенда SDK 1.1 составляет до 64 Кбайт. Таким образом, характеристики данного устройства достаточно для реализации достаточно несложных алгоритмов, в том числе и задача оценки местоположения мобильного объекта.

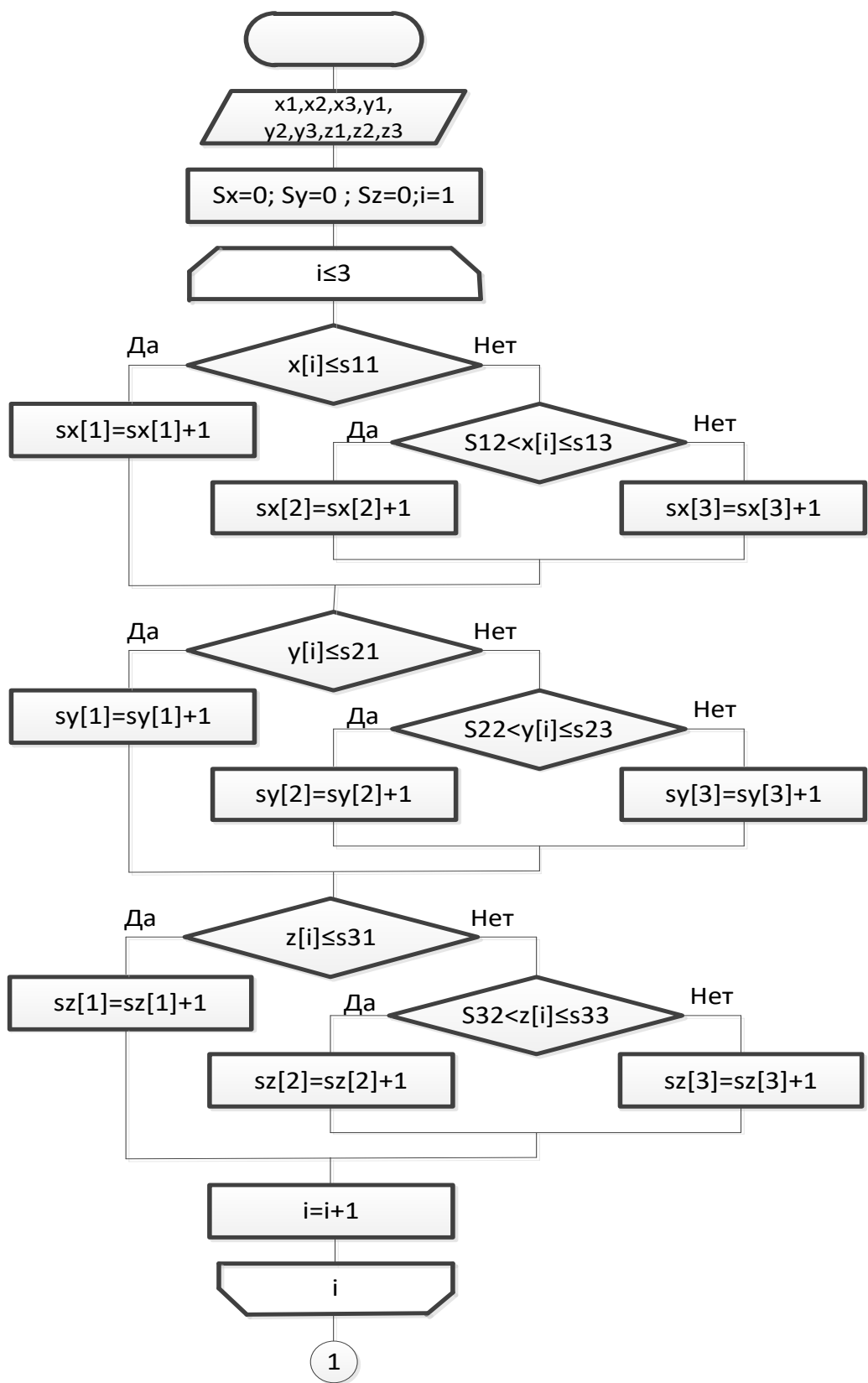


Рисунок 8.1 – Схема алгоритма оценки положения мобильного объекта

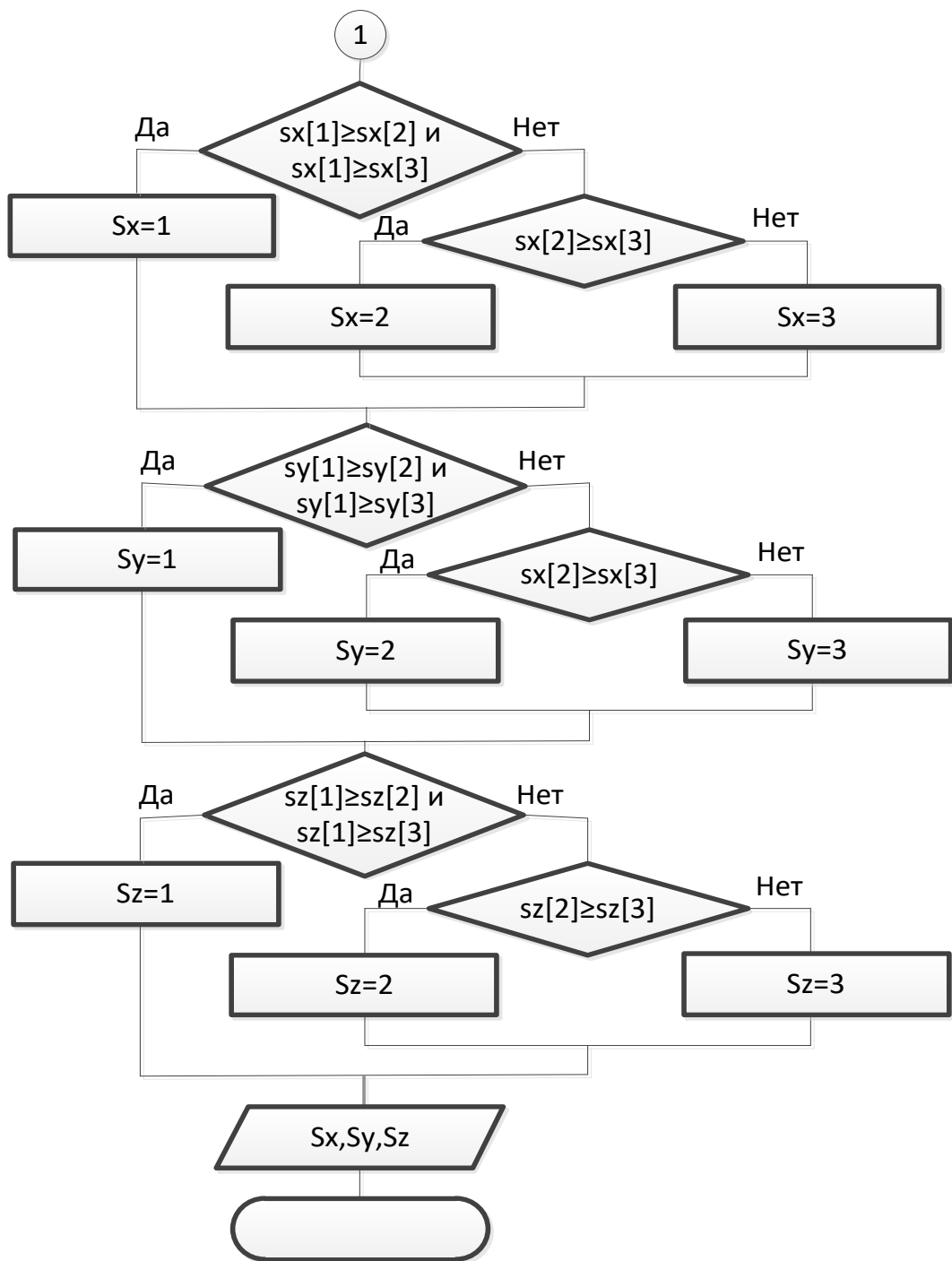


Рисунок 8.2 – Схема алгоритма оценки положения мобильного объекта (продолжение)

На рисунке 8.3 представлена структурная схема системы синтеза программного кода микроконтроллера для решения задачи оценки местоположения МО.

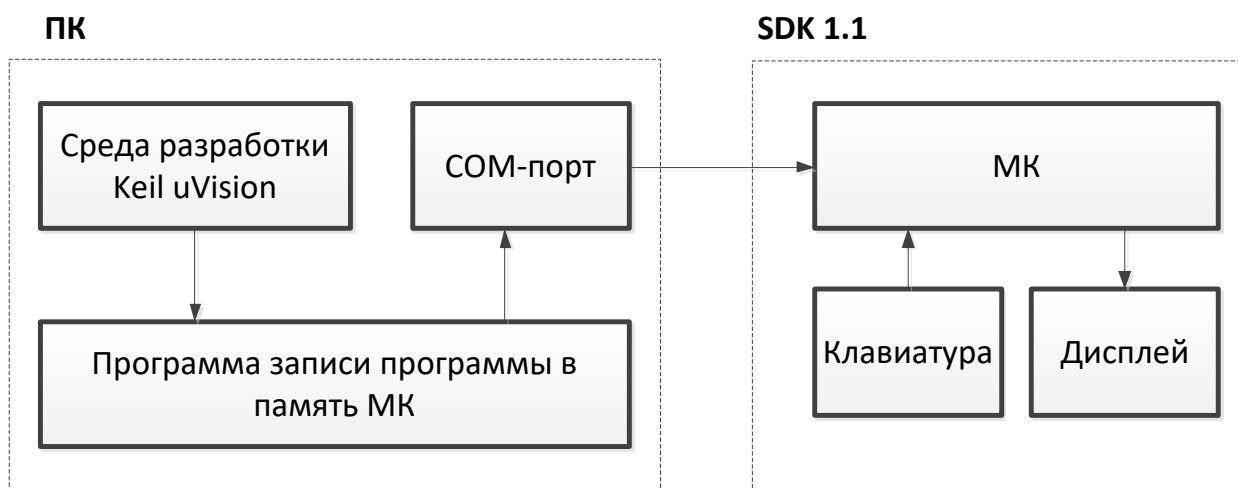


Рисунок 8.3 – Структурная схема системы программирования МК

Исходный код разрабатывается и компилируется в среде разработки Keil uVision C51. Поддерживается программирование на ассемблере и языке С. В результате компиляции формируется HEX-файл с программой, который загружается в память микроконтроллера при помощи утилиты T2 по COM-порту. В работе для имитации сигналов от навигационного оборудования используется клавиатура МК, а для вывода результата - дисплей микроконтроллера.

8.2 Задание на лабораторную работу

1 Изучить алгоритм метода оценки положения МО при помощи микроконтроллера.

2 В интегрированной среде Keil uVision на языке С модифицировать программный код для изменения диапазонов зон по варианту. Программный код предоставляется преподавателем.

3 Провести эксперимент по оценке местоположения МО при помощи микроконтроллера.

4 По результатам работы сделать выводы и составить отчет.

5 Ответить на контрольные вопросы.

8.3 Варианты заданий

Варианты заданий для выполнения лабораторной работы представлены в таблице 8.1.

Таблица 8.1 – Варианты заданий для выполнения лабораторной работы

Номер варианта	Диапазоны изменения координат								
	Зона S1			Зона S2			Зона S3		
	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ
1	0-3	0-2	0-3	2-7	2-5	2-7	6-9	4-8	6-9
2	0-5	0-4	0-5	4-7	3-6	4-7	6-8	5-9	6-8
3	0-2	0-3	0-2	2-6	2-4	2-6	5-9	3-7	5-9
4	0-5	0-5	0-5	4-6	4-7	4-6	5-9	6-9	5-9
5	0-3	0-2	0-3	2-4	2-6	2-4	3-8	5-8	3-8
6	0-4	0-4	0-4	3-5	3-7	3-5	4-9	6-8	4-9
7	0-2	0-5	0-2	2-5	4-6	2-5	4-8	4-9	4-8
8	0-5	0-3	0-5	4-7	2-4	4-7	6-9	3-8	6-9
9	0-4	0-2	0-4	3-6	2-7	3-6	5-8	5-8	5-8
10	0-3	0-4	0-3	2-6	3-5	2-6	5-9	5-9	5-9

8.4 Порядок выполнения работы

1 Запустите интегрированную среду «Keil uVision». В основном меню выберите пункт «Project | Open Project». В появившемся диалоговом окне выберите проект Lr8 (рисунок 8.4).

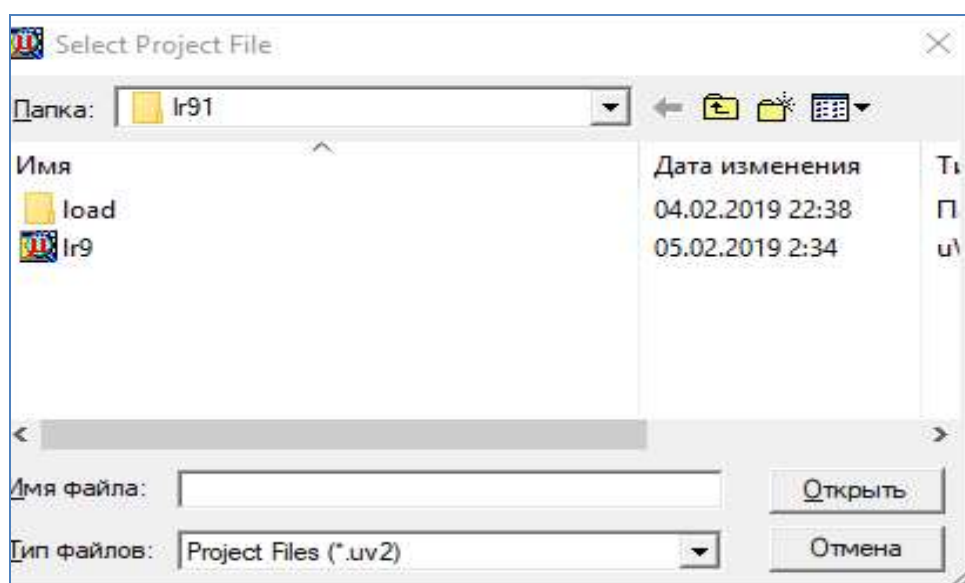


Рисунок 8.4 – Окно выбора проекта

2 В коде программы измените диапазоны состояний по соответствующим осям. На рисунке 8.5 выделен участок кода, где нужно произвести изменения.

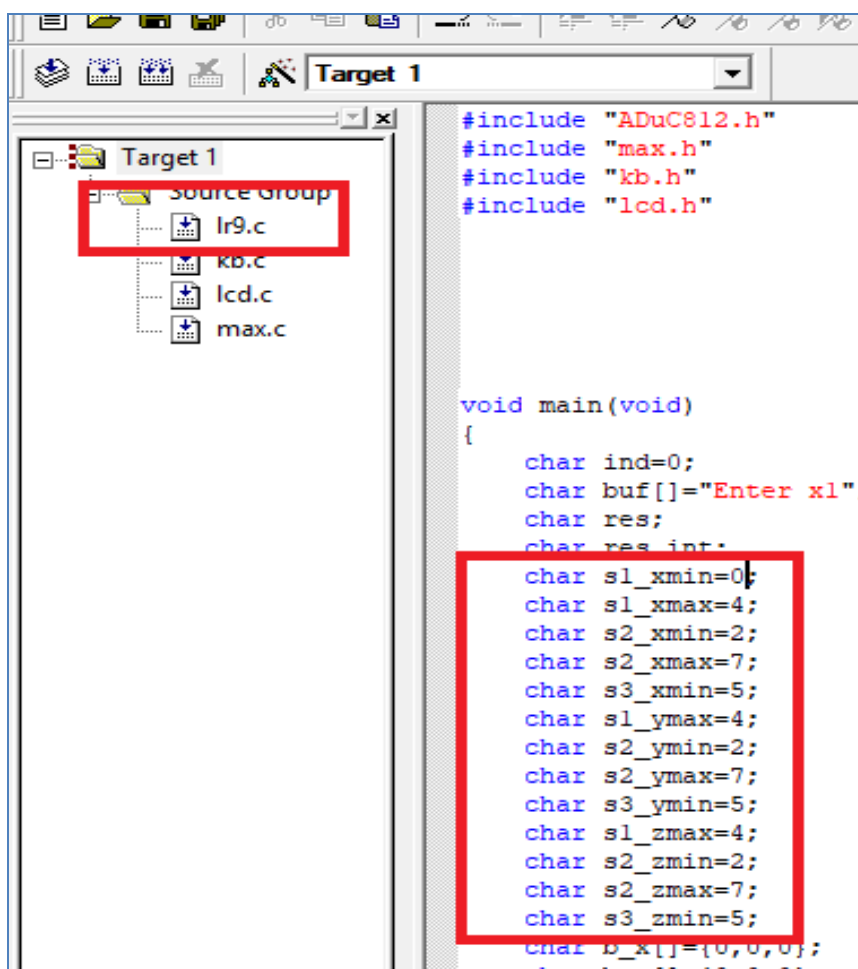


Рисунок 8.5 – Участок программного кода

3 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)».

4 Измените текст конфигурационного файла load.t2 название hex-файла и номер COM-порта:

```
0x2100 0x0 addhexstart Lr9.hex
opencom1
loadhex Lr9.hex
0 term
```

bye

Номера COM-портов, установленных на компьютере, можно посмотреть в «Диспетчер устройств» в разделе порты (рисунок 8.6).

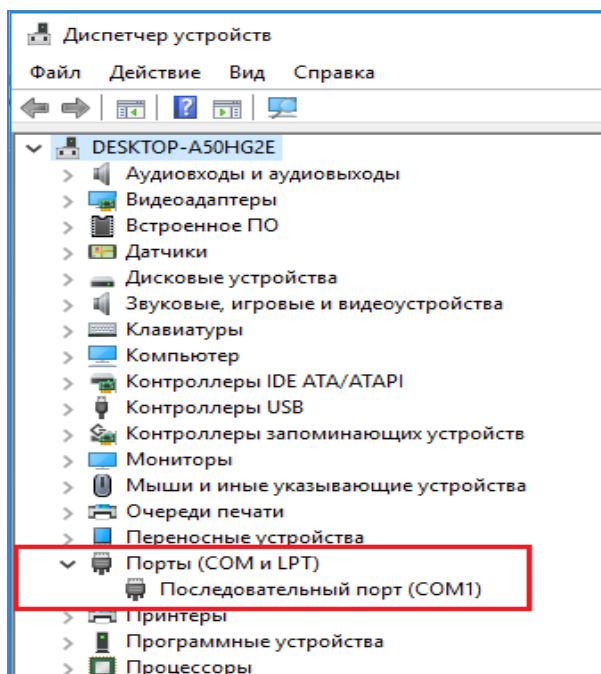


Рисунок 8.6 – Окно программы «Диспетчер устройств»

5 Подключите стенд к компьютеру через COM-порт, включите питание. Запустите на компьютере пакетный файл run_t2.bat и дождитесь загрузки программы в память МК (рисунок 8.7).

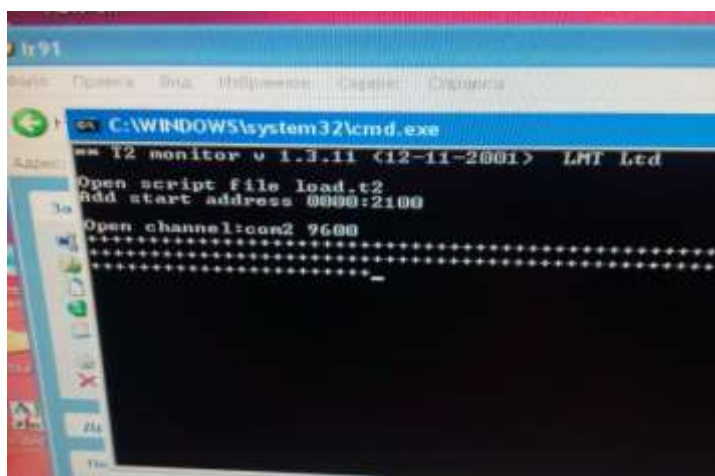


Рисунок 8.7 – Отображение загрузки программы в память учебного стенда на экране персонального компьютера

6 Введите координаты МО через клавиатуру, используя символы от '0' до '9' (один символ – одно значение координаты). Сначала вводятся три координаты по оси X, затем по Y и после по оси Z. Проверьте корректность работы программы (рисунок 8.8).



Рисунок 8.8 – Результат программной реализации метода оценки положения МО в трехмерном пространстве.

7 Составьте отчет по работе и ответьте на контрольные вопросы

8.5 Контрольные вопросы

- 1 Объясните алгоритм определения местоположения МО.
- 2 Какие модули учебного стенда и для чего используются в проведении натурального эксперимента?
- 3 Назовите этапы программирования стенда.
- 4 Для чего необходим файл формата «HEX-80»?
- 5 Как осуществляется загрузка программы в память стенда?
- 6 Что является результатом программной реализации метода оценки положения МО в трехмерном пространстве?

9 Лабораторная работа № 9 (4 часа). Контроль команд оператора в системе управления объектом.

Цель работы: освоить метод контроля команд оператора в системе управления объектом на примере устройства для мониторинга поведения пользователя.

9.1 Теоретическая часть

К исходным данным задачи контроля команд оператора отнесены:

– множество Q контролируемых транзакций, выполняемых пользователем:

$$Q = \{q_1, q_2, \dots, q_j, \dots, q_N\};$$

– множество P информативных признаков: $P = \{p_1, p_2, \dots, p_i, \dots, p_M\};$

– код транзакции K ;

– номер операции в транзакции NO ;

– код операции транзакции KOT .

Под транзакцией в работе понимается кортеж команд для выполнения некоторой технологической операции.

Математическая модель мониторинга поведения пользователя описывает процедуру контроля действий пользователя, представляемую параметрами: K , NO , KOT в конкретный момент времени t . При этом совокупный признак $P(t)$ имеет следующий вид [1Д]:

$$P(t) = (K_NO_KOT)_t.$$

В каждый момент времени t совокупный признак ассоциируется (\equiv) с кодом соответствующей легитимной (правомерной) операции $\langle KOT \rangle_t$, хранящейся в ассоциативной памяти (АП):

$$(K_NO_KOT)_t \equiv \langle KOT \rangle_t$$

В случае отсутствия ассоциации производится запрет на ввод следующей операции.

Технически контроль поведения реализуется подачей совокупного кода $P(t)$ на адресные входы блока ассоциативной памяти. При наличии ассоциации с выходов

АП поступает код легитимной операции KLO , который сравнивается с кодом операции KOT , набираемым пользователем. При равенстве $KLO=KOT$ санкционируется ввод следующего совокупного признака $P(t+1)$. После выполнения всех операций транзакции режим контроля завершается.

На рисунке 9.1 представлена структурная схема устройства для мониторинга поведения пользователя [5Д].

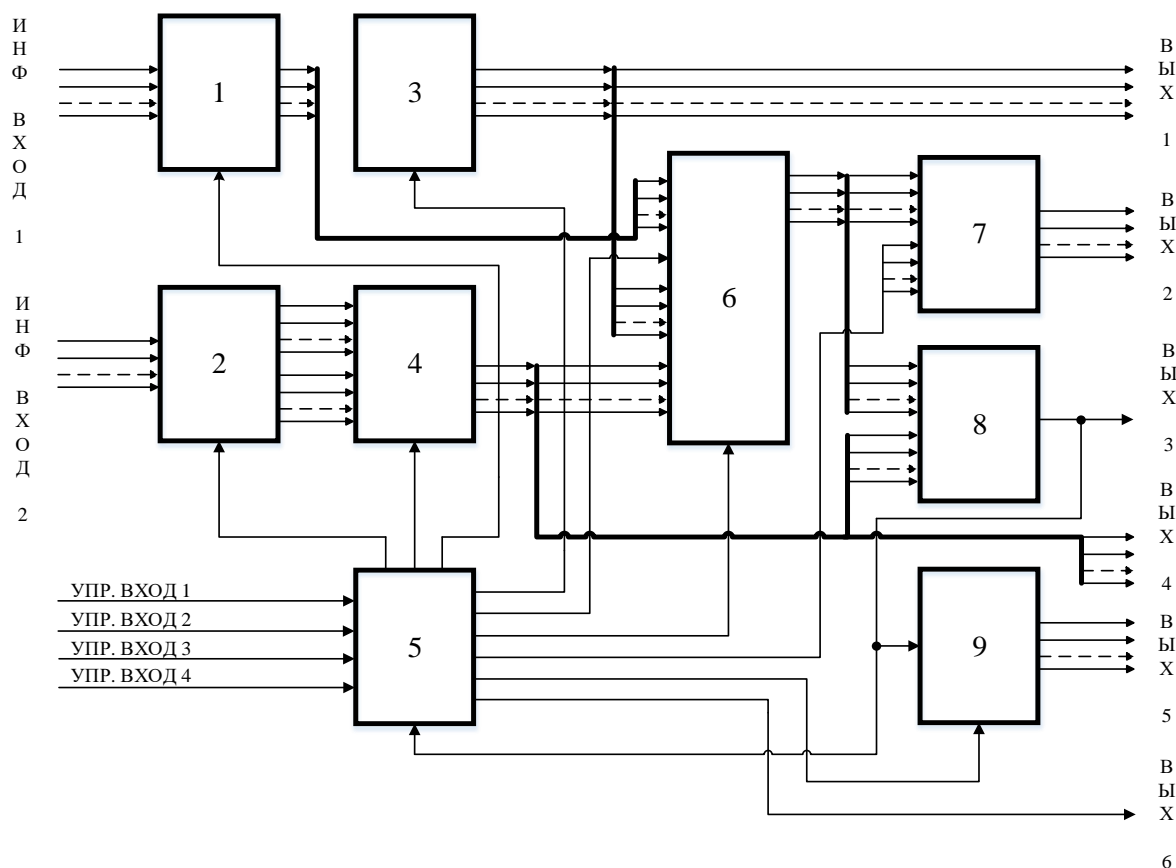


Рисунок 9.1 – Структурная схема устройства для контроля поведения пользователя

Цифрами на схеме обозначены: 1 – регистр кода транзакции, 2 – блок сдвиговых регистров; 3 – счетчик адресов; 4 – многоканальный коммутатор; 5 – блок управления; 6 – блок памяти; 7 – блок логических элементов И; 8 – схема сравнения; 9 – сдвиговый регистр результатов сравнения.

Устройство для контроля поведения пользователя функционирует в трех режимах:

- режим контроля поведения пользователя;
- режим анализа поведения;
- режим обучения.

В режиме контроля на информационные входы устройства 1 и 2 подаются, соответственно, код транзакции и код команды. В блоке памяти, при этом, содержится информация, соответствующая санкционированной последовательности команд. При этом код каждой команды, поданной пользователем, инициирует чтение соответствующего кода из блока памяти. При этом на входы блока сравнения поступает код первой команды пользователя и контрольные коды этой команды из блока памяти. При неравенстве этих кодов режим контроля завершается. При равенстве кодов разрешается ввод последующей команды, по достижении конечного числа команд режим контроля завершается.

Инициализация режима анализа осуществляется путем подачи команды управления на соответствующий управляющий вход устройства. Режим анализа поведения пользователя предназначен для выявления характера ошибок в поведении пользователя при выполнении транзакции. Режим анализа поведения работает аналогично режиму контроля за исключением того, что при возникновении ошибки ввод команд пользователем продолжается. Код корректности команд фиксируется в сдвиговом регистре результатов сравнения, и по достижении конечного числа команд режим анализа завершается.

Режим обучения предназначен для ознакомления пользователя с составом команд транзакции и порядком их задания. Исходное состояние устройства в режиме обучения соответствует конечному состоянию его в режиме анализа. Ознакомление с командами транзакции осуществляется путем считывания контрольных команд из блока памяти независимо от кодов, сформированных в адресной части блока памяти.

Адресный код блока памяти на рисунке 9.2 состоит из 4 групп двоичных разрядов: **A**, **B**, **C** и **D**. Первая адресная группа **A** содержит код транзакции. Код **B** – одноразрядный, он определяет содержание блока памяти для конкретного режима работы устройства, причем значение разряда равно единице, если устройство

работает в режиме контроля и анализа. Значение разряда равно нулю, если устройство работает в режиме обучения. Группа **C** определяет номер операции в транзакции, а группа **D** – код операции транзакции, вводимой пользователем.

9.2 Порядок выполнения работы

На рисунке 9.2 представлена экранная форма имитационной модели работы устройства для контроля поведения пользователя в соответствии с рисунком 1. Таблица модели представлена на листе 1 табличного процессора и состоит из двух частей: верхняя часть представляет работу устройства с данными в десятичной форме представления чисел, нижняя – с данными в двоичной форме.

	A	B	C	D	E	F	G	H
1	Номер	Команда	Адресная	часть	памяти	в	Содержимое	H
2	команды			десятич				
3	в транзакции		A	B	C	D	E	
4	1	К	1	1	1	138	138	
5	2	О	1	2	2	142	142	
6	3	Ж	1	3	3	134	134	
7	4	З	1	4	4	87	87	
8	5	Г	1	5	5	83	83	
9	6	С	1	6	6	145	145	
10	7	Ф	1	7	7	148	148	
11	8			8	8			
12	9			9	0			
13	10			10	0			
14			В ДВОИЧ		НОМ	КОДЕ		
15	1	К	0001	0001	0000	10001010	10001010	
16	2	О	0001	0010	0000	10001110	10001110	
17	3	Ж	0001	0011	0000	10000110	10000110	
18	4	З	0001	0100	0000	01010111	10000111	
19	5	Г	0001	0101	0000	01010011	10000011	
20	6	С	0001	0110	0000	10010001	10010001	
21	7	Ф	0001	0111	0000	10010100	10010100	
22	8			1000	0000			
23	9			1001	0000			РЕЗУЛЬТАТ
24	10			1010	0000			1

Рисунок 9.2 - Экранная форма имитационной модели работы устройства для мониторинга поведения пользователя

На рисунке 9.3 представлена таблица кодов букв русского алфавита, используемая для кодирования команд транзакций, эта таблица сохранена на листе 2 процессора.

	А	В	С	Д	Е	Ф	Г	Н
1	№ п.п	Буква	Десятичный код буквы	Двоичный код буквы	№ п.п.	Буква	Десятичный код буквы	Двоичный код буквы
2	1	А	128	10000000	17	Р	144	10010000
3	2	Б	129	10000001	18	С	145	10010001
4	3	В	130	10000010	19	Т	146	10010010
5	4	Г	131	10000011	20	У	147	10010011
6	5	Д	132	10000100	21	Ф	148	10010100
7	6	Е	133	10000101	22	Х	149	10010101
8	7	Ж	134	10000110	23	Ц	150	10010110
9	8	З	135	10000111	24	Ч	151	10010111
10	9	И	136	10001000	25	Ш	152	10011000
11	10	Й	137	10001001	26	Щ	153	10011001
12	11	К	138	10001010	27	Ъ	154	10011010
13	12	Л	139	10001011	28	Ы	155	10011011
14	13	М	140	10001100	29	Ь	156	10011100
15	14	Н	141	10001101	30	Э	157	10011101
16	15	О	142	10001110	31	Ю	158	10011110
17	16	П	143	10001111	32	Я	159	10011111

Рисунок 9.3 - Экранная форма таблицы кодов букв русского алфавита

Порядок построения и исследования имитационной модели с использованием табличного процессора Excel представлен на экранной форме имитационной модели (рисунок 9.2) и имеет нижеперечисленные пункты.

1 Запустить имитационную программу табличного процессора «Мониторинг поведения пользователя»;

2 В режиме обучения модели для контроля транзакции в соответствии с данными варианта задания сформировать экранное пространство табличного процессора. При этом в колонке **А** выставляются порядковые номера команд в транзакции (до 9 команд включительно). Транзакция задается в виде последовательности букв русского алфавита. Каждая буква соответствует начальной букве названия команды.

3 В колонке **В** задаются начальные буквы русского алфавита, соответствующие начальным буквам команд транзакции, например: **К, О, Ж, З, Г,**

С, Ф;

4 В колонке **С** выставляется код номера транзакции для всех команд, например, 1;

5 В колонку **D** заносятся коды номеров команд в транзакции;

6 В колонку **Е** автоматически заносятся результаты контроля правильности ввода текущей команды: 1, 2, 3 и т.д. Колонка **F** предназначена для ручного ввода кодов команд пользователем в процессе его работы. В колонках: **С, D, Е, F** формируются составные коды адресов кодов истинных команд.

7 В колонку **G** пользователем последовательно заносятся коды истинных команд транзакции, выбираемые пользователем из таблицы кодов команд в соответствии с вариантом задания. Эти коды имитируют содержимое памяти устройства, из которого выбираются команды для сравнения с командами пользователя в процессе контроля. Для чистоты эксперимента цвет шрифта кодов в колонке **G** принимается белым (невидимым). Для адекватного представления работы устройства в нижней части таблицы имитационной модели автоматически производится преобразование всех числовых данных верхней таблицы в двоичный код. На этом пункте завершается процесс подготовки (обучения) модели для контроля поведения пользователя при вводе транзакции с номером 1.

8 Для контроля корректности управления пользователем последовательно по порядку вводятся все коды команд транзакции в колонке **F**. При корректном вводе каждого кода в колонке **Е** появляется номер последующей команды. При корректном вводе последней команды в этой колонке появляется номер команды на единицу больше, свидетельствующий об успешном завершении ввода транзакции. При корректном вводе всех команд в ячейке **H24** появляется единичный код корректного ввода, либо нулевой код ошибочного ввода команд.

9 Провести анализ корректности задания команд пользователем.

9.3 Задание на лабораторную работу

1 Ознакомиться с теоретической частью работы

2 В соответствии с данными варианта работы построить имитационную

модель в среде табличного процессора Excel с использованием программы «Мониторинг поведения пользователя».

- 3 Произвести ввод заданной транзакции в колонке **F**.
- 4 Провести анализ корректности ввода транзакции.
- 5 Составить отчет по лабораторной работе, в котором указать:
 - тему и цель работы;
 - экранную форму имитационной модели;
 - вывод по проделанной работе в соответствии с поставленной целью.

9.4 Варианты заданий

Выбор последовательности команд в транзакции определяется заглавными буквами фамилии студента по журналу группы.

9.5 Контрольные вопросы

- 1 Дайте определение имитационной модели.
- 2 Дайте определение процессу мониторинга поведения пользователя.
- 3 Объясните принцип работы устройства для контроля поведения пользователя.
- 4 Объясните принцип заполнения колонки **E** в имитационной модели мониторинга.
- 5 Объясните принцип заполнения ячейки (логическую функцию) результата **H24** модели.
- 6 Приведите примеры приложений рассмотренного метода контроля поведения пользователя.
- 7 Какие требования предъявляются к имитационной модели объекта.
- 8 8. Укажите достоинства и недостатки имитационного моделирования в среде табличного процессора.
- 9 Объясните математическую модель мониторинга поведения пользователя.

10 Литература, рекомендуемая для изучения тем

10.1 Основная литература

1. Абрамова, Т.В., Метод оперативного прогнозирования и ранжирования рисков информационной безопасности на основе ассоциативного подхода / Т.В. Абрамова, Т.З. Аралбаев, Г.Г. Аралбаева, Р.Р. Галимов // Вопросы развития современной науки и практики в период становления цифровой экономики: Материалы международной научно-практической конференции (18 октября 2018 г.). – Санкт-Петербург: Санкт-Петербургский государственный лесотехнический университет имени С.М. Кирова. 2018. URL: http://www.konferenc.ru/konferenc10_10_18_3.html (дата обращения 10.11.2018).

2. Абрамова, Т.В., программное средство: Ранжирование рисков от угроз на основе ассоциативного принципа / Абрамова, Т.З. Аралбаев, Р.Р. Галимов, Е.В. Каменева // Свидетельство о регистрации электронного ресурса. Рег. № 1646. Дата регистрации: 01.10.2018. Оренбургский государственный университет. Университетский фонд электронных ресурсов.

3. Аралбаев Т. З. Методики начального изучения беспилотного мобильного объекта как объекта информатизации в условиях кафедры вуза/ Т. З. Аралбаев, Т. В. Абрамова, Р. Р. Галимов, Е. В. Каменева, Г. А. Халин// Университетский комплекс как региональный центр образования, науки и культуры: материалов Всероссийской научно-методической конференции; Оренбург: ОГУ, 2019. – 8 с.

4. Аралбаев Т. З. Применение макросов табличного процессора в задаче изучения имитационной модели мониторинга перемещения мобильного объекта в трехмерном пространстве/ Т. З. Аралбаев, Р. Р. Галимов, Е. В. Каменева, Т. В. Абрамова, Е. Ю. Захаров// Университетский комплекс как региональный центр образования, науки и культуры: материалов Всероссийской научно-методической конференции; Оренбург: ОГУ, 2019. – 7с.

5. Учебный стенд SDK-1.1. Руководство пользователя. [Электронный ресурс]. – Режим доступа: http://embedded.ifmo.ru/sdk/sdk11/doc/sdk11_userm_v1_0_11.pdf

6. Lazarus: бесплатная среда по разработке программного обеспечения // Lazarus-rus [Электронный ресурс]. - URL: <https://lazarus-rus.ru/> - (дата обращения 31.01.2019).

10.2 Дополнительная литература

1. Аралбаев, Т. З. Оптимизация методов контроля технического состояния распределенных автоматизированных систем в условиях воздействия пространственно-временных угроз на основе мониторинга сетевых информационных потоков: монография / Т. З. Аралбаев, Г. Г. Аралбаева, Т. В. Абрамова, Р. Р. Галимов, А. В. Манжосов; Оренбургский гос. ун-т. - Оренбург: ОГУ, 2018. -160 с. ISBN – 978-5-7410-2202-3.

2. Горюнов, А.Г. Архитектура микроконтроллера intel 8051: учеб. пособие./ А.Г. Горюнов, С.Н. Ливенцов. Томск: Изд-во ТПУ, 2005. - 86 с.

3. Отладочное средство для GPS/GLONASS модулей SIM68хю. Режим доступа: <http://mt-system.ru/catalog/besprovodnye-resheniya/gps-i-gpsglonass-moduli/gps-i-gpsglonass-moduli-simcom/otladochnye-i->

4. Павлов, А.Н., Соколов Б.В. Принятие решений в условиях нечеткой информации / А.Н.Павлов, Б.В. Соколов / учебное пособие./ С-Петербург, 2006. - 72 с.

5. Патент № 2675896. Российская Федерация. Устройство для контроля поведения пользователя / Абрамова Т.В., Аралбаев Т.З., Каскинов И.И., Хатеев М.Д. Заявка № 2018100997. Приоритет изобретения 10.01.2018. Оpubл. в БИ № 36, 2018 г.

6. Протокол обмена IEC 61162-1 (NMEA-0183). Режим доступа: https://www.irz.ru/uploads/files/226_1.pdf.

7. ARSDK Protocols. [Электронный ресурс]. – Режим доступа: https://developer.parrot.com/docs/bebop/ARSDK_Protocols.pdf

8. Arsdk-xml. [Электронный ресурс]. – Режим доступа: <https://github.com/Parrot-Developers/arsdk-xml/blob/master/xml/ardrone3.xml>

9. Network Manager - Network Tools & Utilities. [Электронный ресурс]. – Режим доступа: <http://4pda.ru/forum/index.php?showtopic=931360>.

Приложение А

(обязательное)

Листинг программы

```
Name Lr2
; указание адресов регистров специальных назначений
SV EQU 7h
DPP EQU 84h
T2CON EQU 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
```