

Министерство образования и науки Российской Федерации
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Инфокоммуникационных технологий»

ПРОЕКТ ПРОВЕРЕН
Рецензент

_____ Хашимов А.Б.

“_25_” ____декабря__ 2018_ г.

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ИКТ

_____ С.Н. Даровских

“_14_” ____января__ 2019_ г.

Аппаратно-программное обеспечение для имитатора ультразвуковой РЛС

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОМУ КВАЛИФИКАЦИОННОМУ ПРОЕКТУ (ВКП)
ЮУрГУ – Д.11.05.01.2019.307.00 ПЗ ВКП

Консультанты:

По экономике

_____ Р.Ш. Закиров

“_12_” ____декабря__ 2018_ г.

Руководитель проекта

старший инженер отдела ТСО

ООО НПФ «Учтех-Профи»

_____ К.С. Новиков

“_25_” ____декабря__ 2018_ г.

По охране труда

_____ И.С. Окраинская

“_21_” ____декабря__ 2018_ г.

Автор работы

студент группы КЭ-637

_____ И.Г. Шафигин

“_12_” ____декабря__ 2018_ г.

Нормоконтролер

_____ В.Д. Спицына

“_11_” ____января__ 2019_ г.

АННОТАЦИЯ

Шафигин И.Г. Аппаратно-программное обеспечение для имитатора ультразвуковой РЛС – Челябинск: ЮУрГУ, ВШ ЭКН, 2019, 101 с., 33 рис., 7 табл., 24 источника.

В данной дипломном проекте рассматривается процесс разработки управляющего программного обеспечения для стенда имитации ультразвуковой радиолокационной станции. Основными объектами исследования являются управляющий микроконтроллер, процесс генерации, приема и обработки принятых УЗ волн, и отображение результатов на экране персонального компьютера посредством программы, написанной на языке высокого уровня *Kotlin*.

Проектирование и разработка цифровой части стенда осуществляются на основании технической документации на предыдущую разработку – учебно-лабораторный комплекс «РЛС-02-16».

В схемотехническом и конструкторском разделах приводятся описание стенда, а также графическое объемное их изображение.

В программном разделе приводятся процесс разработки программного обеспечения для персонального компьютера и аппаратного программного обеспечения. В приложениях имеются блок схемы алгоритма управляющего контроллера и программный код приемной части персонального компьютера.

Также рассматриваются вопросы экономической эффективности разработки программного обеспечения. В разделе БЖД приводятся основные нормативные документы и правила безопасности при эксплуатации стенда.

В заключительной части сделаны выводы по эффективности разработки и оценены перспективы ее дальнейшего развития.

					<i>ЮУрГУ – Д.11.05.01.2019.307.00 ПЗ</i>			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.	Шафигин И.Г.				Аппаратно-программное обеспечение для имитатора ультразвуковой РЛС	Лит.	Лист	Листов
Провер.	Новиков К.С.					л	2	101
Реценз.						ЮУрГУ Кафедра ИКТ		
Н. Контр.	Спицына В.Д.							
Утверд.	Даровских							

ультразвуковой РЛС

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	6
1 Анализ технического задания	8
2 Принцип действия РЛС	10
2.1 Принцип формирования и управления диаграммой направленности	10
2.2 Амплитудный метод измерения дальности	11
2.3 Частотный метод определения дальности	12
2.4 Фазовый метод определения дальности	14
2.5 Амплитудный метод измерения угловых координат	15
2.6 Фазовый метод определения угловых координат	20
2.7 Доплеровский метод определения скорости	22
3 Анализ конструкции учебно-лабораторного комплекса	24
3.1 Блок визуализации и внешнего управления	25
3.2 Радиолокационная стойка	26
3.3 Рама с поворотным подвесом	31
3.4 Комплектация	33
4 Анализ функциональной схемы устройства	34
5 Программный раздел	37
5.1 Разработка программы для микроконтроллера	37
5.2 Разработка программы для компьютера	40
6 Организационно-экономический раздел	48
6.1 Понятие параметров сетевого планирования	48
6.2 Анализ этапов разработки, построение сетевого графика	50
6.3 Расчет разработки ПО	56

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

АФАР – активная фазированная антенная решетка;

АФД – амплитудно-фазовый детектор;

АЦП – аналого-цифровой преобразователь;

БППУ – блок передачи, приема, управления, обработки, сопряжения;

ДН – диаграмма направленности;

МКП – метод критического пути;

МФУ – многофункциональное устройство;

ОС – операционная система;

ПК – портативный компьютер;

ПО – программное обеспечение;

ПЭВМ – портативная электронно-вычислительная машина;

РЛС – радиолокационная станция;

ФАР – фазированная антенная решетка;

ЭМ – электромагнитный;

JVM – Java Virtual Machine – виртуальная машина Java;

SPI – Serial Peripheral Interface – последовательный периферийный интерфейс.

						Лист
					ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	5
Изм.	Лист	№ докум.	Подпись	Дата		

ВВЕДЕНИЕ

НПП «Учтех-Профи» является признанным лидером российских производителей учебной техники. Выпускаемое учебное оборудование соответствует всем международным требованиям качества продукции. Разработки стендов осуществляются под научным руководством и при участии ведущих педагогов и ученых ЮУрГУ (НИУ) и других базовых вузов и техникумов России.

Учебно-лабораторный комплекс «РЛС-02-16» является разработкой НПП «Учтех-Профи». В комплекс входит стенд, который представляет собой имитатор радиолокационной станции, где вместо электромагнитных (ЭМ) волн используются акустические, ультразвуковые волны, длина которых составляет 8,575 мм. Основные характеристики и внешний вид стенда детально рассмотрен в разделе анализа конструкции лабораторного комплекса данного проекта.

Основным достоинством комплекса является то, что распространение, отражение, преломление, сложение, вычитание и другие волновые эффекты для ЭМ и акустических волн одинаковы. Следовательно, с точки зрения волновых эффектов, использование акустической волны полностью соответствует ЭМ волне.

Стоит заметить, что наиболее коротковолновые радиолокационные станции используют длину волны порядка 3 см. Разрешающая способность РЛС зависит от длины волны, чем меньше длина волны, тем более мелкие объекты можно обнаружить. Использование длины волны 8,5 мм позволяет смоделировать и оценить работу реальной РЛС, спроектированной на используемую длину волны, что на данный момент невозможно для ЭМ волн [1], [2].

Если считать, что работа акустической РЛС происходит в реальном времени, то объект, удаленный от неё на 1 м, соответствует реальному объекту, удаленному на 874,6 км (~ 1000 км).

						Лист
					ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	6
Изм.	Лист	№ докум.	Подпись	Дата		

Также стоит заметить, что для выбранной длины волны акустических колебаний присутствует затухание, также, как и для электромагнитных. Это затухание следует учитывать как при использовании ЭМ волн, так и при использовании акустических волн. Затухание акустических волн на используемой длине волны составляет около 0,3 дБ/м. Эти потери следует учитывать при расчетах.

Всё вышперечисленное делает актуальной разработку и дальнейшее развитие рассматриваемого учебно-лабораторного комплекса, так как высокий уровень знаний достигается не только теоретическим багажом обучающихся, но и уровнем практических навыков.

					ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

1 Анализ технического задания

Основной задачей данного дипломного проекта является разработка программного обеспечения аппаратной части, а также компьютерной части стенда имитатора ультразвуковой РЛС.

Главной целью разработки является усовершенствование текущей версии программного обеспечения учебного стенда, создания нового кода и его оптимизация.

Согласно техническому заданию, компьютерная часть программного обеспечения должна выполняться в операционной системе Astra Linux, которая является дистрибутивом GNU/Linux.

Для разработки ПО был выбран язык программирования Kotlin, среда разработки – IntelliJ IDEA.

Kotlin – статически типизированный язык программирования, работающий поверх JVM (Java Virtual Machine). Язык был разработан компанией JetBrains и на данный момент он поддерживается и активно развивается мировым сообществом [3]. Основные преимущества Kotlin перед другими языками программирования:

– программы, написанные на этом языке, работают поверх виртуальной машины Java. Язык имеет полную, прямую и обратную совместимость с Java;

– программы являются кроссплатформенными – могут выполняться на любой операционной системе, поддерживающей JVM. Astra Linux поддерживает JVM;

– язык имеет более высокий уровень, чем C/C++: он типобезопасный, то есть имеется встроенная проверка на null значения, нет возможности напрямую обращаться к оперативной памяти, но имеется встроенный в JVM «сборщик мусора», позволяющий не заботиться о созданных и не используемых объектах классов;

– лаконичный синтаксис – позволяет писать программы кратко и понятно.

									Лист
									8
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ				

IntelliJ IDEA – интегрированная среда разработки ПО, разработанная компанией JetBrains, поэтому наиболее полно поддерживает работу с языком программирования Kotlin.

Разрабатываемое программное обеспечение предполагает пользовательский интерфейс для ввода и отображения требуемой информации. Для реализации пользовательского интерфейса будет использована JavaFX – платформа на основе Java, в которую входят многочисленные элементы пользовательского интерфейса.

ПК связывается с учебный стендом посредством USB интерфейса. Для работы с USB в разрабатываемой программе используется стандартная реализация доступа к USB устройствам в языке Java.

					ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

2 Принцип действия РЛС

В реальных РЛС при радиолокационном наблюдении информация об объектах переносится радиосигналами, которыми являются электромагнитные колебания. В исследуемом учебно-лабораторном комплексе в качестве радиолокационных сигналов используются акустические волны. Как было описано во введении данного проекта, все эффекты, необходимые для радиолокации, присущие ЭМ колебаниям, также полностью соответствуют акустическим колебаниям.

В акустической (ультразвуковой) РЛС используется метод активной радиолокации [4], основанный на излучении ультразвуковых волн и приеме отраженных от целей волн приемным устройством.

2.1 Принцип формирования и управления диаграммой направленности

Антенный блок РЛС, который подробно описан в разделе анализа конструкции лабораторного комплекса, состоит из целой системы излучателей, представляющей фазированную антенную решетку (ФАР). Поэтому далее будем рассматривать линейную ФАР.

Рассмотрим линейную ФАР, состоящую из n одинаковых излучателей, расположенных по координатной оси y . Для расчета ДН линейной ФАР в дальней зоне используется выражение [5], [6]:

$$F(\varphi) = F_{\text{изл}}(\varphi) \sum_{m=1}^n I_m \exp(i\psi_m) \exp(iky_m \sin \varphi), \quad (2.1)$$

где $F(\varphi)$ – комплексная ДН в плоскости азимутальных углов φ (горизонтальная плоскость);

$F_{\text{изл}}(\varphi)$ – ДН единичного излучателя ФАР;

I_m и ψ_m – амплитуда и фаза возбуждения излучателя с номером m , соответственно;

y_m – координата расположения излучателя.

2.2 Амплитудный метод измерения дальности

Измерение дальности целей основано на измерении времени запаздывания t_D радиолокационных сигналов.

При амплитудном (импульсном) методе измерения измеряется время характерного изменения амплитуды принимаемого радиосигнала. Часто используемым видом модуляции является импульсная модуляция.

Рассмотрим устройство импульсной дальномерной РЛС. Функциональная схема изображена на рисунке 2.1.

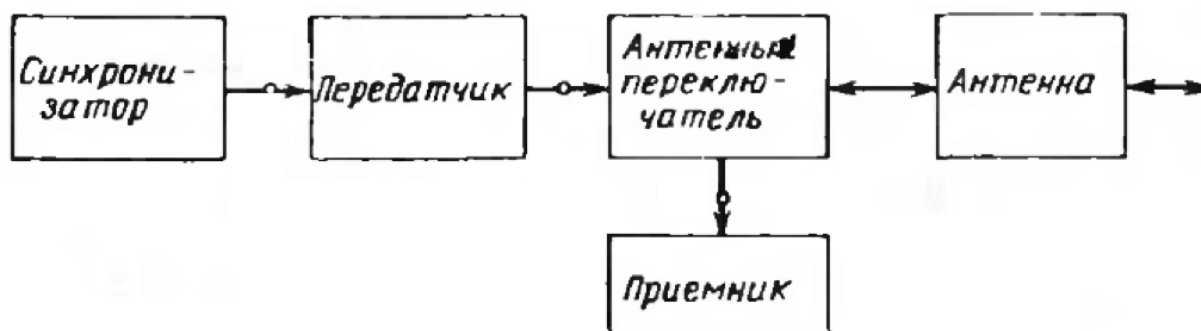


Рисунок 2.1 – Функциональная схема импульсного измерителя дальности

Передатчик генерирует импульсы длительностью $\tau_{и}$ с периодом повторения $T_{и}$. Антенный переключатель подсоединяет антенну к передатчику на время генерации импульса и к приемнику на все остальное время. Отраженные импульсные сигналы запаздывают на время t_D [7].

К достоинствам импульсного дальномера можно отнести: возможность построения РЛС с одной антенной, удобство одновременного измерения дальности многих целей. К недостаткам: необходимость использовать значительных импульсных мощностей, сложности при измерении малых дальностей.

Функциональная схема дальномера, реализованного в учебно-лабораторном комплексе, изображена на рисунке 2.2.

Рассмотрим ее подробнее.

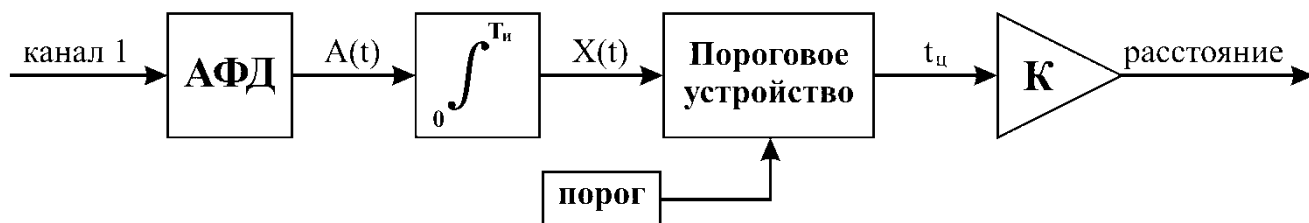


Рисунок 2.2 – Функциональная схема измерителя дальности в учебном комплексе

Передающая часть формирует радиоимпульс длительностью 1 мс, частотой заполнения 40 кГц. Одновременно с началом передачи происходит запись сигнала приемником (приемной антенной). Запись продолжается в течение 25 мс, что соответствует прохождению звука в воздухе на расстояние 8,575 м. Соответственно, максимальное расстояние, на котором работает РЛС около 4,3 м.

2.3 Частотный метод определения дальности

Данный метод основан на использовании частотной модуляции непрерывных колебаний, время запаздывания измеряется путем определения разности частот излученных колебаний и отраженного сигнала. Функциональная схема изображена на рисунке 2.3.

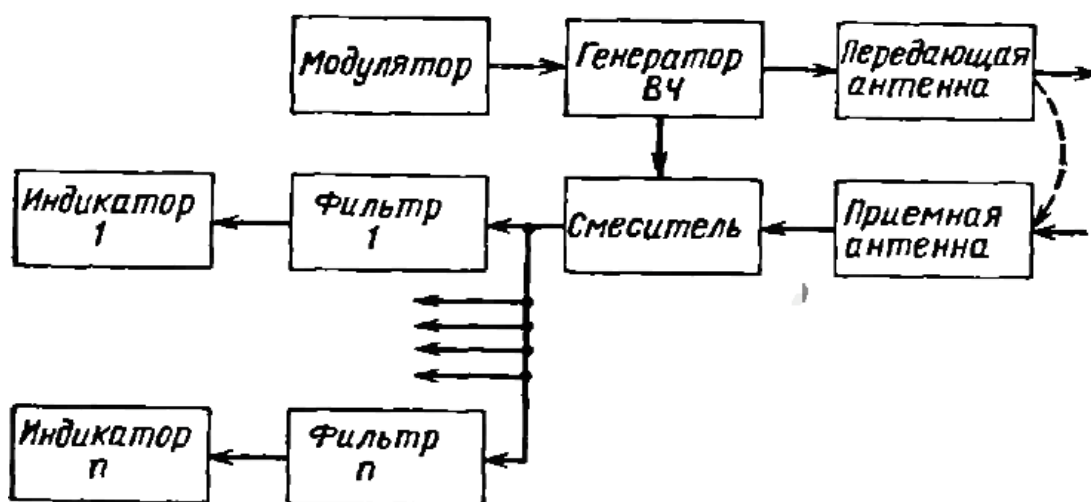


Рисунок 2.3 – Функциональная схема дальномера с частотной модуляцией

Генератор высокочастотных колебаний, управляемый модулятором, вырабатывает сигнал с частотой, изменяющейся по периодическому закону.

Частота сигнала, отраженного от неподвижной цели, будет меняться по такому же закону, но только со сдвигом во временной оси на время запаздывания t_D . Отраженные сигналы и колебания генератора подводятся к смесителю. Образующаяся на выходе смесителя разностная частота пропорциональна дальности цели. Если круговая частота излучения будет определяться по формуле (2.2)

$$\omega_{\text{изл}} = \omega_0 + \frac{\Delta\omega_M}{T_M} t, \quad (2.2)$$

где $\Delta\omega_M$ – девиация частоты передатчика, то частота сигнала, отраженного от неподвижной цели, будет равна

$$\omega_{\text{отр}} = \omega_0 + \frac{\Delta\omega_M}{T_M} \left(t - \frac{2D}{c} \right). \quad (2.3)$$

Расстояние до цели будет рассчитываться по формуле (2.4)

$$D = \frac{c f_p}{2 \Delta f_M F_M}. \quad (2.4)$$

К достоинствам данного метода можно отнести: возможность измерять очень малые дальности, малая мощность излучения. К недостаткам: необходимость разделять излучаемые и принимаемые колебания, высокие требования к линейности изменения частоты.

В учебном комплексе РЛС при измерении расстояния частотным методом работает следующим образом. Функциональную схему можно увидеть на рисунке 2.4.

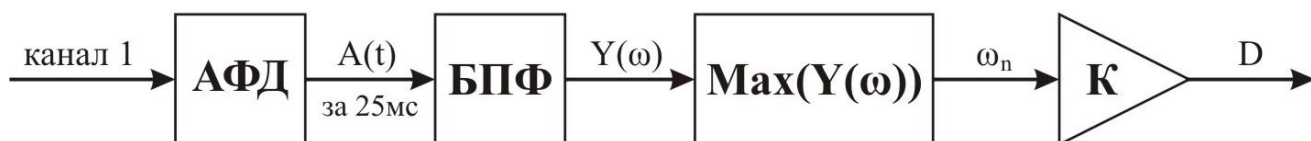


Рисунок 2.4 – Функциональная схема дальномера на частотной модуляции в учебном комплексе

Передающая часть генерирует сигнал с частотой, линейно изменяющейся во времени с 39 кГц до 41 кГц. Время, за которое происходит изменение сигнала – 25 мс, что соответствует прохождению звука в воздухе на расстояние 8,575 м.

Одновременно с началом излучения начинается запись принятого сигнала. Начиная с некоторого момента, на приемник приходит и сигнал, излученный с передатчика и сигнал, отраженный от цели. Разность частот, принятого сигнала пропорциональна дальности до цели, т.к. частота передатчика меняется линейно. Диапазон исследуемых расстояний – 200...4000 мм. Расстояния свыше 3500 мм могут определяться неуверенно.

2.4 Фазовый метод определения дальности

Данный метод основан на измерении разности фаз излученных синусоидальных колебаний и принятых радиосигналов. Функциональная схема простейшего фазового дальномера изображена на рисунке 2.5.

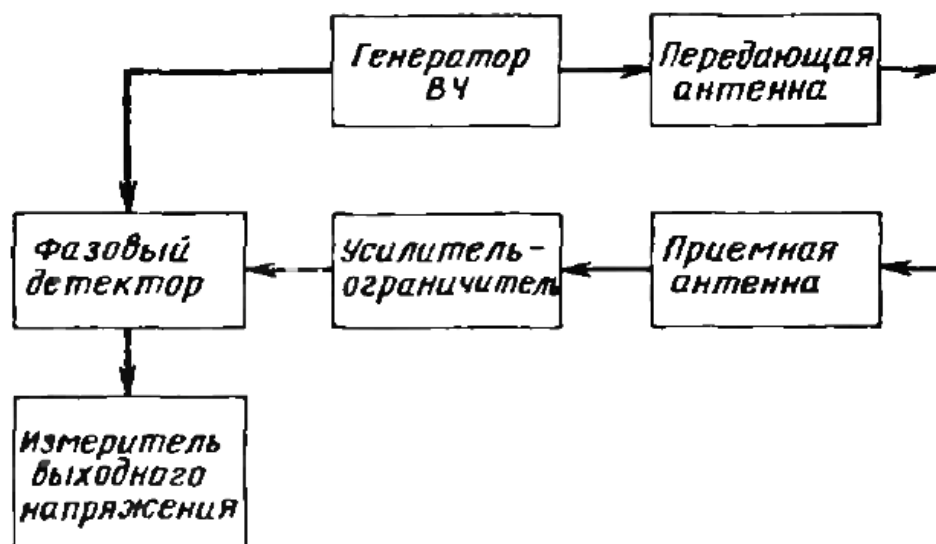


Рисунок 2.5 – Функциональная схема простейшего фазового измерителя дальности

Генератор создает незатухающие колебания частоты ω_0 . Фаза колебаний определяется по формуле (2.5):

$$\psi_{\text{изл}} = \omega_0 t + \psi_1, \quad (2.5)$$

где ψ_1 – начальное значение фазы.

Фаза принимаемого сигнала определяется по формуле (2.6):

$$\psi_{\text{пр}} = \omega_0(t - t_D) + \psi_{\text{отр}} + \psi_{\text{РЛС}} + \psi_1, \quad (2.6)$$

где $\psi_{\text{отр}}$ – фазовый сдвиг, связанный с отражением радиоволны,

$\psi_{\text{РЛС}}$ – фазовый сдвиг в цепях РЛС (известен, так как поддается измерениям).

Принятые колебания сравниваются с колебаниями генератора, разность фаз рассчитывается по формуле (2.7)

$$\Delta\psi = \frac{4\pi}{\lambda} D - \psi_{\text{отр}} - \psi_{\text{РЛС}}. \quad (2.7)$$

В учебном комплексе фазовый метод определения дальности реализован следующим образом.

Измеряется расстояние до цели амплитудным методом, находится некоторое расстояние R . Это расстояние может быть выражено через длину волны, используемую в РСЛ по формуле (2.8):

$$R = N \cdot \lambda + \Delta \cdot \lambda, \quad (2.8)$$

где N – количество целых длин волн от РЛС до цели,

Δ – дробная часть длины волны, от 0 до 1.

Расстояние рассматривается как некоторое целое число длин волн $N \cdot \lambda$ и $\Delta \cdot \lambda$. Δ соответствует изменению фазы принятого сигнала от 0 до 2π .

Плюсом данного метода можно считать следующее: если принимаемая отраженная фаза стабильна в некотором диапазоне, например, $\varphi = 1250 \pm 30$, то можно говорить о точности измерения $\pm 3/360 \cdot \lambda = \pm 0.0083 \cdot \lambda = 71.1725$ мкм.

С другой стороны, при отражении волн от реальных целей фаза является величиной случайной, и не может быть использована для точного измерения расстояния.

2.5 Амплитудный метод измерения угловых координат

В настоящий время широко используются несколько вариантов амплитудного метода: максимума, минимума, сравнения и равносильный.

Рассмотрим каждый из методов.

При пеленгации по *методу максимума* плавно изменяется угловое положение антенны, в течении этого времени она принимает сигнал от цели.

									Лист
									15
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ				

Отсчет угловой координаты цели производится в тот момент, когда амплитуда сигнала достигает максимального значения. Функциональную схему угломерного устройства, основанном на методе максимума можно увидеть на рисунке 2.6.

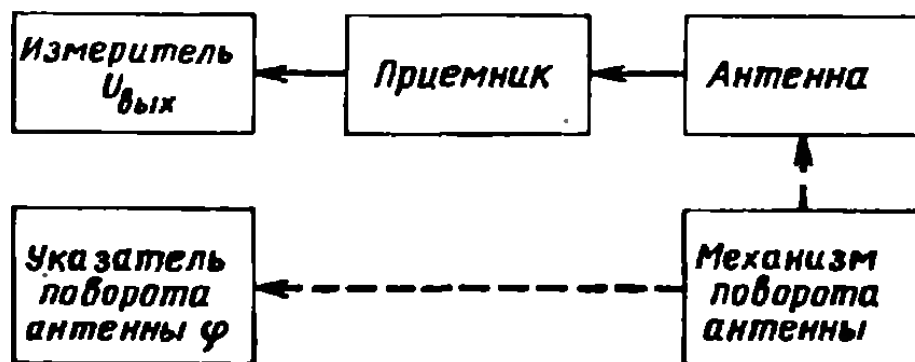


Рисунок 2.6 – Функциональная схема угломерного устройства с отсчетом по максимуму

Механизм поворота вращает антенну, одновременно приводится в действие указатель поворота антенны, который отсчитывает направление поворота антенны.

Когда ось антенны совпадает с направлением на цель, выходное напряжение приемника достигает максимума. В этот момент указатель покажет пеленг цели. Пеленгационная характеристика данного метода изображена на рисунке 2.7. Она определяется по формуле (2.9)

$$U_{\text{вых}}(\varphi) = kF(\varphi), \quad (2.9)$$

где k – коэффициент пропорциональности,

$F(\varphi)$ – диаграмма направленности антенны.

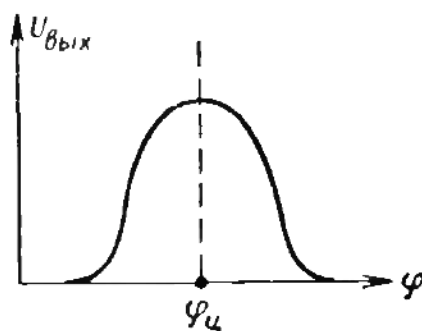


Рисунок 2.7 – Пеленгационная характеристика при пеленгации по максимуму

Достоинством рассматриваемого метода является простота его реализации. Основной недостаток — это относительно низкая точность измерений угловых координат.

Метод минимума отличается от метода максимума тем, что отсчет угловой координаты производится в момент уменьшения до минимума выходного напряжения приемника. Диаграмма направленности должна иметь в средней части провал. Получить такую диаграмму направленности можно используя две антенны, повернутые в пространстве относительно друг друга на равный угол. Пеленгационная характеристика изображена на рисунке 2.8.

Данный метод может обеспечивать относительно высокая точность измерений, но низкая амплитуда сигнала вблизи направления пеленга. Метод получил более широкое применение в радионавигации.

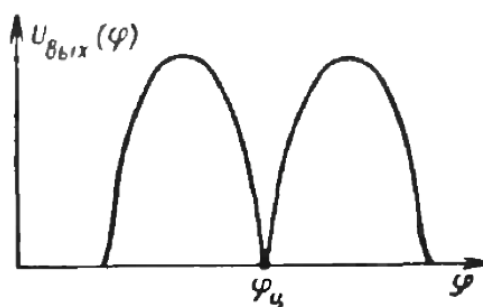


Рисунок 2.8 – Пеленгационная характеристика при пеленгации по минимуму

Метод сравнения характеризуется тем, что пеленг цели определяется по соотношению амплитуд сигналов, принятых одновременно двумя антеннами. Функциональная схема пеленгационного устройства приведена на рисунке 2.9. В счетнорешающей схеме происходит сравнение сигналов. Наиболее просто осуществляется вычитание одного сигнала из другого.

Выходное напряжение зависит от абсолютных амплитуд сигналов и, следовательно, будет изменяться в зависимости от расстояния до цели, отражающих свойства цели, поглощения в среде и т.д.

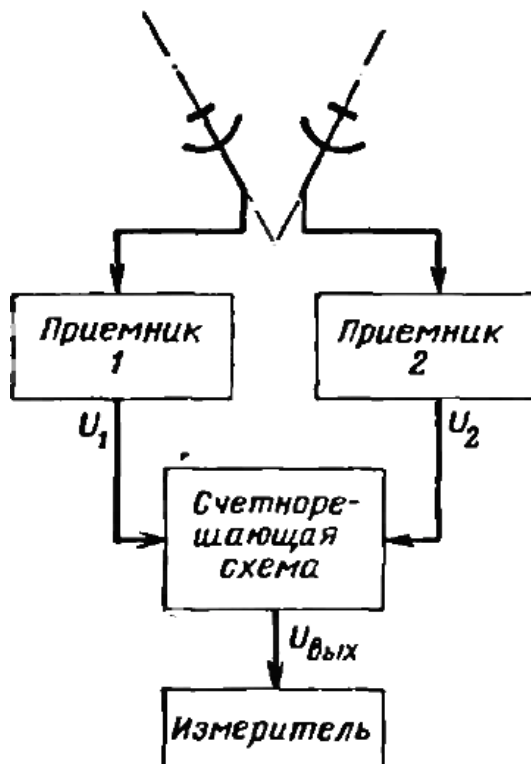


Рисунок 2.9 – Функциональная схема пеленгатора, в котором используется метод сравнения.

Вид пеленгационной характеристики изображен на рисунке 2.10. Она определяется по формуле (2.10).

$$U_{\text{ВЫХ}}(\varphi) = \ln \frac{k_1 F_1(\varphi_{\text{ц}})}{k_2 F_2(\varphi_{\text{ц}})}, \quad (2.10)$$

где k_1, k_2 – коэффициенты пропорциональности первой и второй антенны соответственно,

$F_1(\varphi_{\text{ц}}), F_2(\varphi_{\text{ц}})$ – диаграмма направленности в направлении цели для первой и второй антенны соответственно.

Основным достоинством метода сравнения является возможность мгновенного определения направления на цель в пределах широкого сектора при неподвижной антенной системе, при этом этому методу свойственна низкая точность измерений.

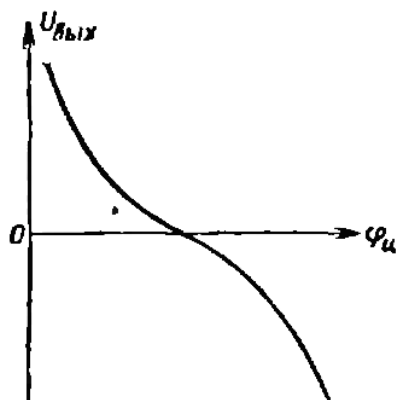


Рисунок 2.10 – Пеленгационная характеристика при использовании метода сравнения

Частным методом сравнения является *равносигнальный метод* пеленгации. Он также основан на сравнении амплитуд сигналов, принимаемых антеннами, но для отсчета пеленга добиваются равенства сигналов. Угловые координаты цели определяются по углу поворота антенны.

Равносигнальный метод характеризуется высокой точностью, так как при измерении используется небольшой участок диаграммы направленности с относительно большой крутизной.

В учебно-лабораторном комплексе для измерения угла амплитудным методом используется метод максимума из-за его простоты и достаточной точности. Функциональная схема измерителя угловых координат амплитудным методом изображена на рисунке 2.11.

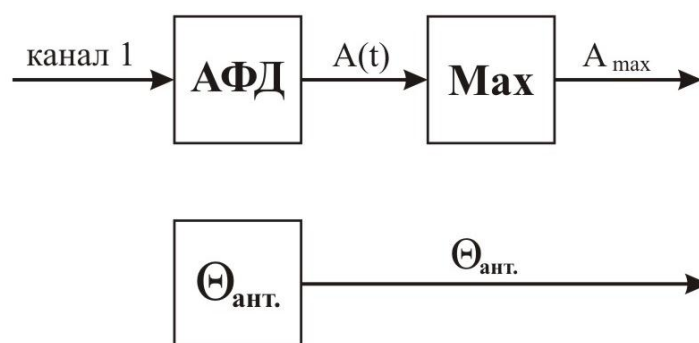


Рисунок 2.11 – Функциональная схема работы измерителя угловых координат в учебно-лабораторном комплексе

2.6 Фазовый метод определения угловых координат

Фазовые методы основаны на измерении разности фаз сигналов, принимаемых различными антеннами. Функциональная схема фазового угломера изображена на рисунке 2.12.

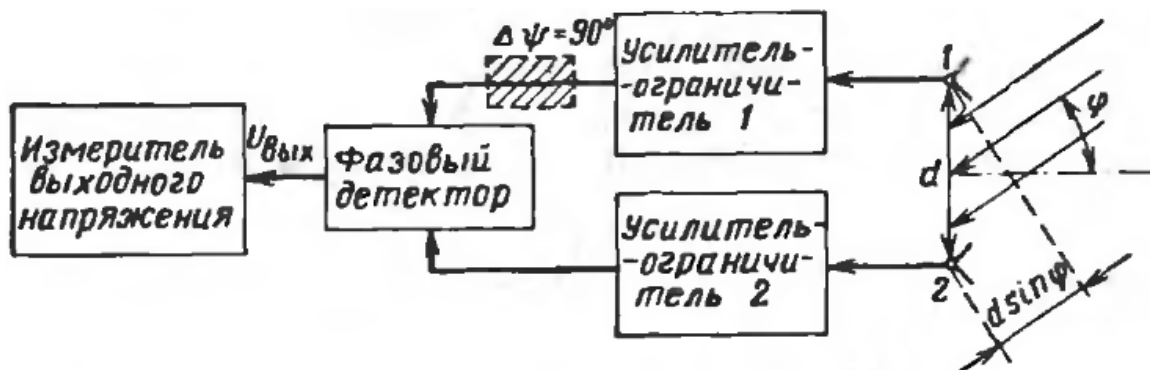


Рисунок 2.12 – Функциональная схема фазового измерителя угловых координат

Пусть в точках 1 и 2 расположены две приемные антенны, расстояние между которыми равно d . Принятые сигналы подводят к фазовому детектору. Выходное напряжение фазового детектора определяется только разностью фаз колебаний, формула (2.11).

$$U_{\text{ВЫХ}} = k \cos \Delta\psi. \quad (2.11)$$

Если направление прихода радиоволны составляет угол φ с перпендикуляром базы, то фазовый сдвиг колебаний в антеннах приближенно равен

$$\Delta\psi = \frac{2\pi d}{\lambda} \varphi. \quad (2.12)$$

С учетом формулы (2.12) пеленгационная характеристика будет

$$U_{\text{ВЫХ}}(\varphi) = k \cos \left(\frac{2\pi d}{\lambda} \varphi \right). \quad (2.13)$$

Она изображена на рисунке 2.13 (кривая 1).

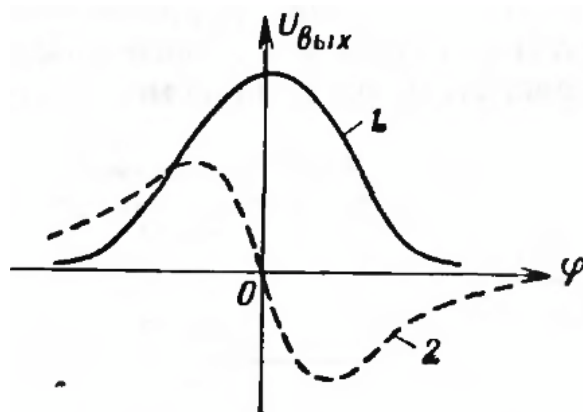


Рисунок 2.13 – Пеленгационная характеристика фазового пеленгатора

Из формулы (2.13) видно, что точность измерения угловых координат вблизи значения $\varphi = 0$ низка. А также нельзя определить направление смещения цели от перпендикуляра к базе. Эти недостатки устраняются введением искусственного фазового сдвига сигнала на 90° . Получим пеленгационную характеристику, соответствующую формуле (2.14), она изображена на рисунке 2.13 (кривая 2).

$$U_{\text{вых}}(\varphi) = k \sin\left(\frac{2\pi d}{\lambda} \varphi\right). \quad (2.14)$$

Фазовый метод характеризуется относительно высокой точностью.

Недостатком является неоднозначность отсчета и отсутствие разрешения целей.

В учебно-лабораторном комплексе реализован модифицированный вариант фазового метода определения угловых координат. Функциональная схема представлена на рисунке 2.14.

Здесь предварительно определяется расстояние до цели амплитудным методом, откуда определяется временной сдвиг сигнала, который за это время доходит до цели. На основе этого временного сдвига и фаз принятых сигналов определяется фазовый сдвиг.

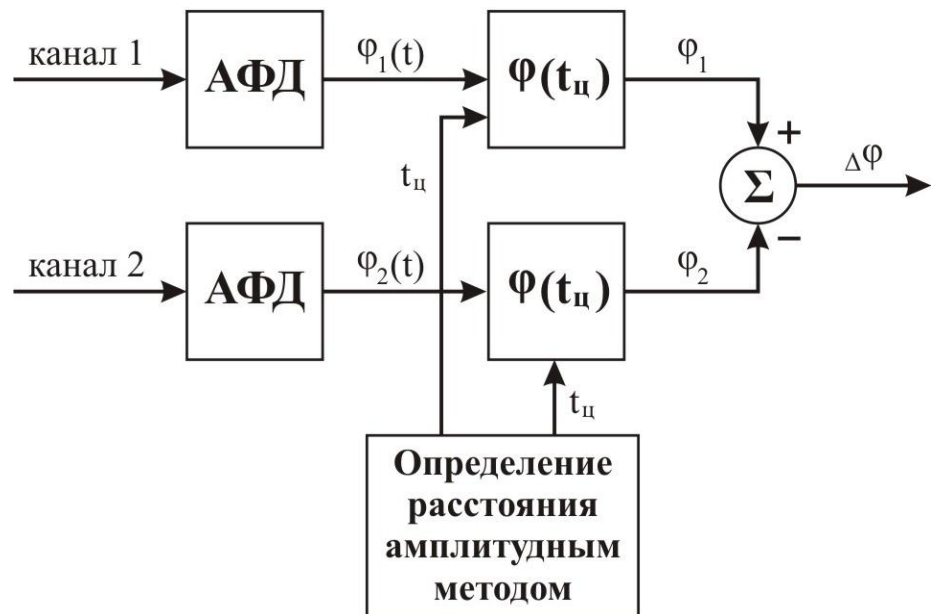


Рисунок 2.14 – Функциональная схема фазового измерителя угловых координат

2.7 Доплеровский метод определения скорости

Измерение скорости основано на определении доплеровского сдвига частоты отражающего сигнала. Схема простейшей станции, измеряющей доплеровскую частоту сигнала, приведена на рисунке 2.15.

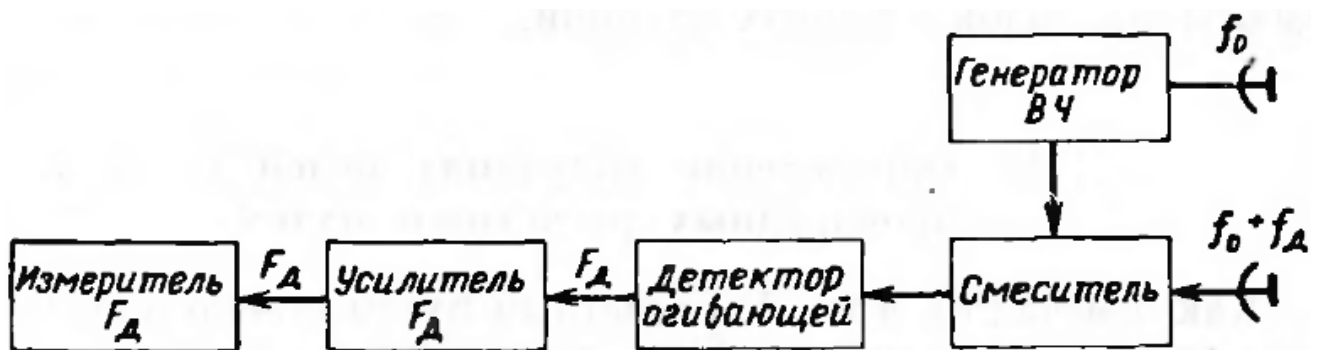


Рисунок 2.15 – Функциональная схема измерителя доплеровской частоты

Генератор высокой частоты (f_0) создает незатухающие колебания, излучаемые в пространство. Частота сигнала, отраженного от движущейся цели, будет отличаться от f_0 на доплеровский сдвиг F_D . Принятые колебания смешиваются с колебаниями генератора. На выходе смесителя образуются колебания с частотой Доплера. Огибающая ВЧ колебаний выделяется с помощью амплитудного детектора, далее колебания на частоте F_D усиливаются и подводятся к измерителю частоты. Скорость цели соответствует формуле (2.15).

$$V_p = \frac{F_d \lambda}{2}. \quad (2.15)$$

К достоинствам метода можно отнести простоту устройства станции, нечувствительность к медленным уходам частоты генератора. К недостаткам: высокие требования к кратковременной стабильности частоты генератора, влияние паразитных модуляций колебаний генератора, возникновение помех при наличии нескольких целей.

Функциональная схема измерителя скорости доплеровским методом, реализованного в учебно-лабораторном комплексе, изображена на рисунке 2.16.

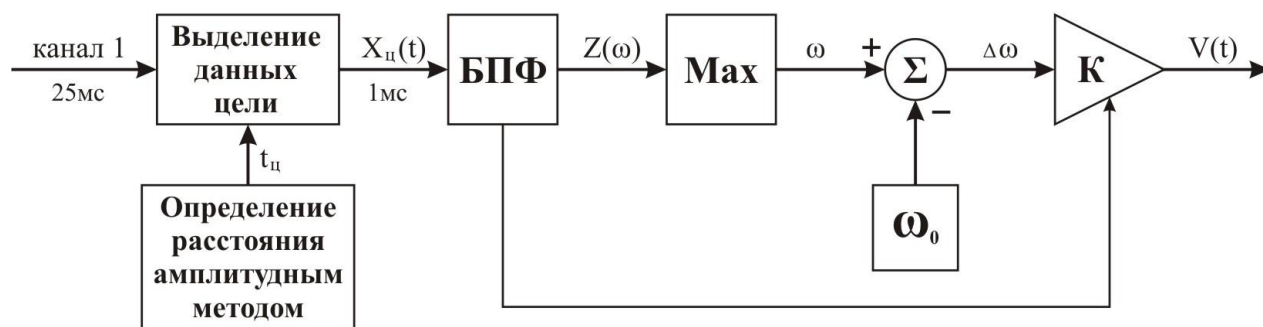


Рисунок 2.16 – Функциональная схема измерителя скорости доплеровским методом

Из приемника сигнал длительностью 25 мс (соответствует расстоянию примерно 8,5 м) проходит обработку, из него выделяются данные цели. Далее новый сигнал разбивается на под сигналы длительностью 1 мс и подвергаются быстрому преобразованию Фурье. Получаем спектр сигнала. Находим частоту с максимальной амплитудой и вычитаем из нее частоту работы генератора. Полученный сигнал с частотой $\Delta\omega = \omega - \omega_0$ умножаем на К. Коэффициент пропорциональности К зависит от длины БПФ. На основе полученной частоты находим скорость движения цели.

3 Анализ конструкции учебно-лабораторного комплекса

Учебно-лабораторный комплекс «Радиолокационные станции обнаружения подвижных объектов на базе АФАР-16» РЛС-02-16 представляет собой аппаратно-программный комплекс, включающий в себя несколько взаимосвязанных блоков, некоторые из которых являются основными, без которых работа с комплексом невозможна, остальные дополнительные.

Общий вид комплекса показан на рисунке 3.1. Он состоит из радиолокационной стойки с антенным блоком, мобильной антенной системы на собственной стойке, рамы с поворотным подвесом и ПЭВМ для управления.

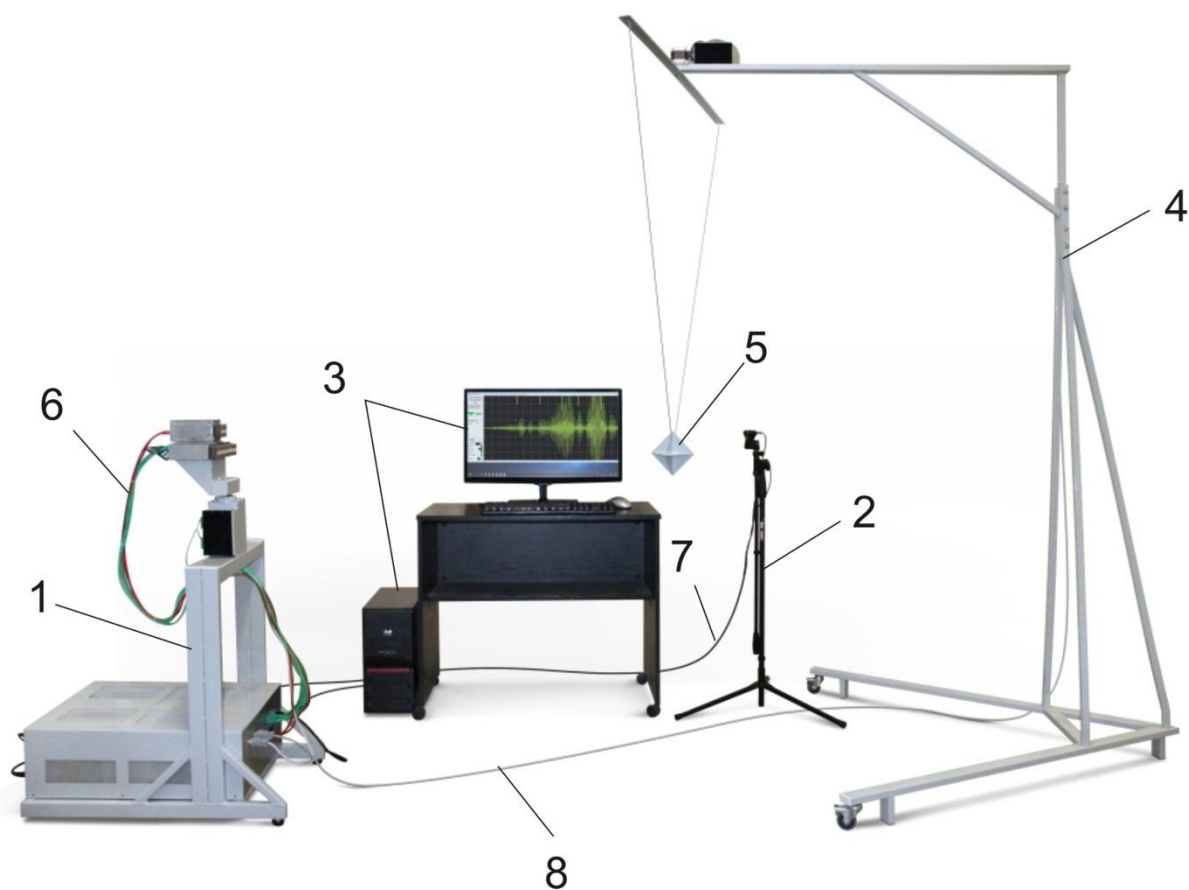


Рисунок 3.1– Общий вид комплекса

Номерами на рисунке 3.1 обозначены:

- 1- радиолокационная стойка;
- 2- мобильная антенная система;
- 3- блок визуализации и внешнего управления (ПК);

Изм.	Лист	№ докум.	Подпись	Дата

- 4- рама с поворотным подвесом для объектов обнаружения;
- 5- цель (объект) обнаружения;
- 6- патч-корды подключения антенного блока к блоку передачи, приема, управления, обработки, сопряжения (БППУ);
- 7- патч-корд с усилителем линии для подключения приемника или излучателя на стойке к БППУ;
- 8- кабель подключения поворотного устройства к БППУ.

Рассмотрим каждый из основных блоков по-отдельности.

3.1 Блок визуализации и внешнего управления

Он представляет системный блок ПК с монитором, клавиатурой и мышью. Монитор выбирается в соответствии с данным требованием: Acer 21,5” (1920 × 1080). Содержание системного блока в соответствии с технической документацией:

- операционная система: Astra Linux;
- процессор: Intel Core i5 (частота 3,2 ГГц, в режиме Turbo 3,6 ГГц);
- материнская плата: Gigabyte GA-H110M-S2LGA;
- оперативная память: DDR4 4 Гб;
- жесткий диск: HDD 500 Гб;
- количество разъемов USB: от 1 шт.

Системный блок выполняется в вертикальном форм-факторе типа MidiTower со стандартными размерами (в мм): 185 - ширина × 410 - высота × 480 - длина. На основной стороне корпуса имеются кнопки включения/выключения и перезагрузки.

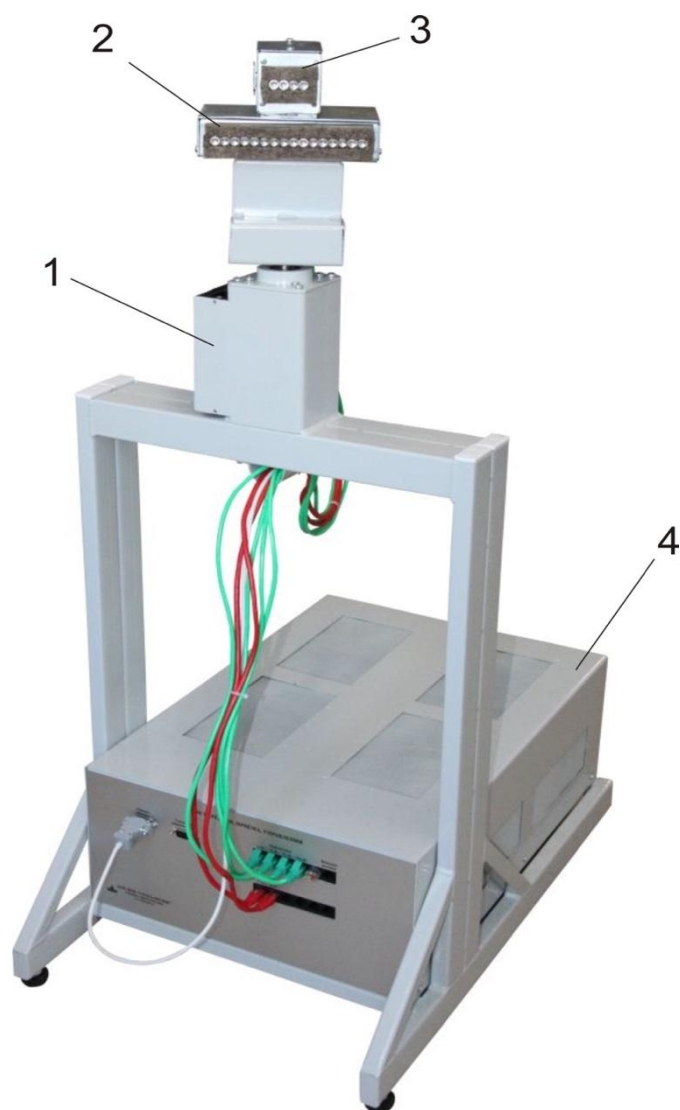
Блок визуализации и внешнего управления обеспечивает:

- передачу данных на блок приема, передачи, обработки и сопряжения по USB-порту;
- прием данных с блока приема, передачи, обработки и сопряжения по USB-порту;

- формирование команд для антенного блока;
- формирование команд для управления поворотным подвесом;
- сбор данных через интерфейс USB от блока записи и обработки принятых сигналов, их трансляцию, хранение, обработку и визуализацию.

3.2 Радиолокационная стойка

На рисунке 3.2 изображена радиолокационная стойка в сборе.



- 1- поворотное устройство;
- 2- антенный блок излучателей (АФАР);
- 3- антенный блок приемников;
- 4- блок передачи, приема, управления, обработки, сопряжения (БППУ).

Рисунок 3.2 – Радиолокационная стойка в сборе

Изм.	Лист	№ докум.	Подпись	Дата

Рассмотрим каждую часть стойки подробнее.

В верхней части радиолокационной стойки, изображенной на рисунке 3.2, находится антенный блок, состоящий из блока приемников и блока излучателей (АФАР). Под ним расположено поворотное устройство, обеспечивающее поворот антенного блока в горизонтальной плоскости влево и вправо на угол 90 градусов от нормали стойки. Разрешение в пределах одного оборота $\pi/1600$ радиан (0,1125 градуса).

Антенный блок со стороны разъемов можно увидеть на рисунке 3.3.



Рисунок 3.3 – Вид антенного блока со стороны разъемов

Номер возле разъема антенного блока обозначает номера подключаемых приемников и излучателей.

Рассмотрим антенный блок излучателей передающей антенны. Вид сверху антенного блока излучателей ФАР изображен на рисунке 3.4.

- передача сигналов управления на поворотное устройство;
- передача накопленных данных от приемника и поворотного устройства на блок визуализации и внешнего управления через интерфейс USB;
- управление углом поворота ФАР;
- управление движением объекта (цели) обнаружения.

БППУ содержит: блок управления движением цели, блок формирования луча и зондирующего импульса, блок усилителей излучателей, блок усилителей приемников, блок сбора и обработки информации с приемников, устройство обмена информацией с ПК.

Блок управления движением цели обеспечивает:

- передачу управляющих команд от компьютера для рамы с поворотным подвесом для объектов обнаружения;
- поворот/вращение объекта обнаружения вокруг собственной оси на 360 градусов с шагом 1,8 градуса;
- поворот/вращение объекта обнаружения по окружности диаметром до 980мм в диапазоне угловых скоростей от 0 до 6 радиан/с.

Блок формирования луча и зондирующего импульса обеспечивает формирование диаграммы направленности и зондирующего импульса с заданными характеристиками:

- когерентного/независимого формирования сигнала по 16 каналам излучения;
- цифровое задание фазы для каждого излучателя;
- цифровое задание амплитуды для каждого излучателя;
- изменение частоты с минимальным шагом 1 Гц.
- блок усилителей излучателей (16 каналов усиления) обеспечивает:
- коэффициент усиления по напряжению 20 дБ;
- значение амплитуды сигналов на выходе 6 В;
- полоса пропускания усилителя 300 кГц на нагрузке 2,4 нФ.

										Лист
										29
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ					

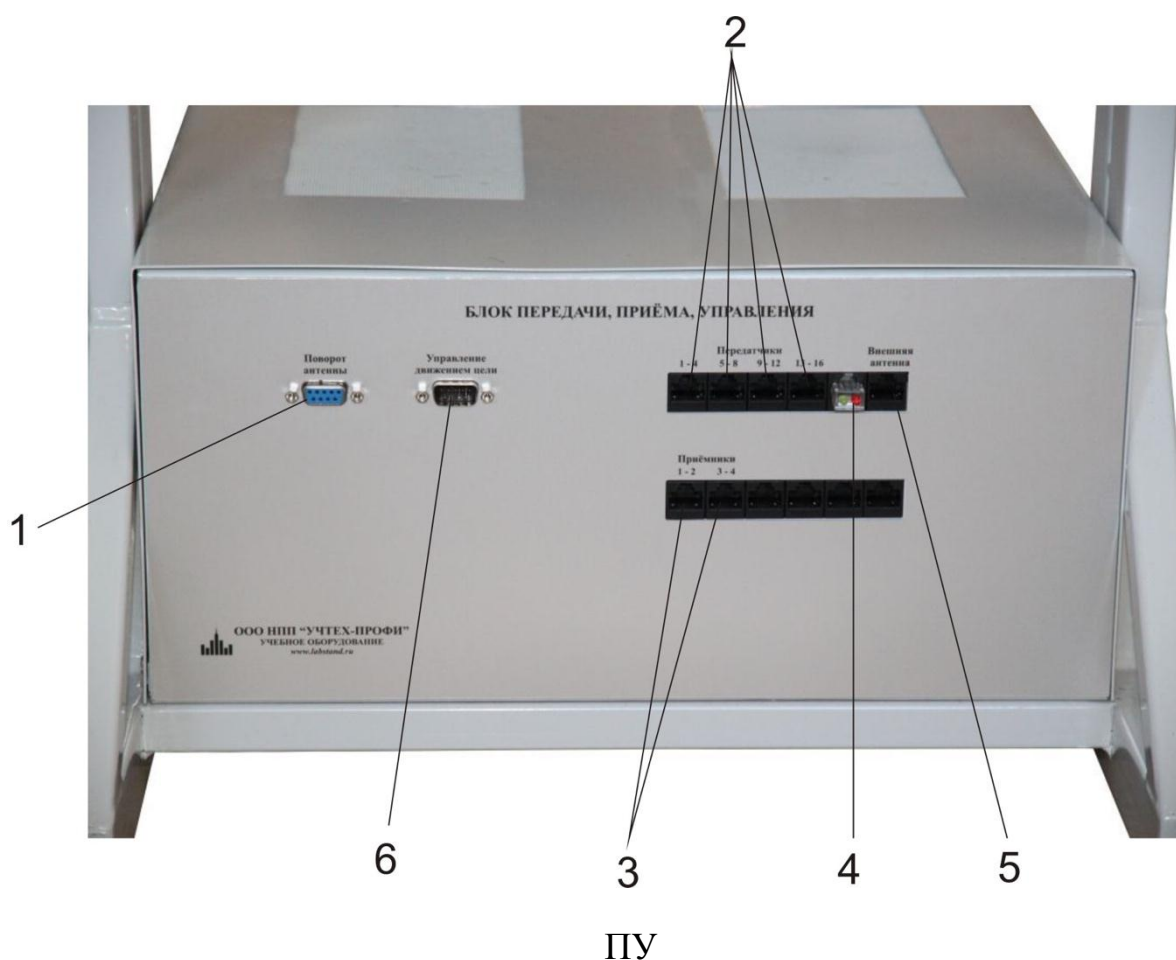
Блок сбора и обработки информации с приемников обеспечивает:

- одновременный сбор информации с нескольких каналов приема;
- передачу данных на компьютер в реальном времени.

Частота опроса каждого канала – 2 МГц (для одного или двух приемников).

Блок усилителей приемников обеспечивает четыре канала логарифмического усиления и один канал линейного усиления. Полоса пропускания усилителей каждого канала 200 кГц.

На рисунках 3.5 и 3.6 изображены лицевая и задняя часть БППУ

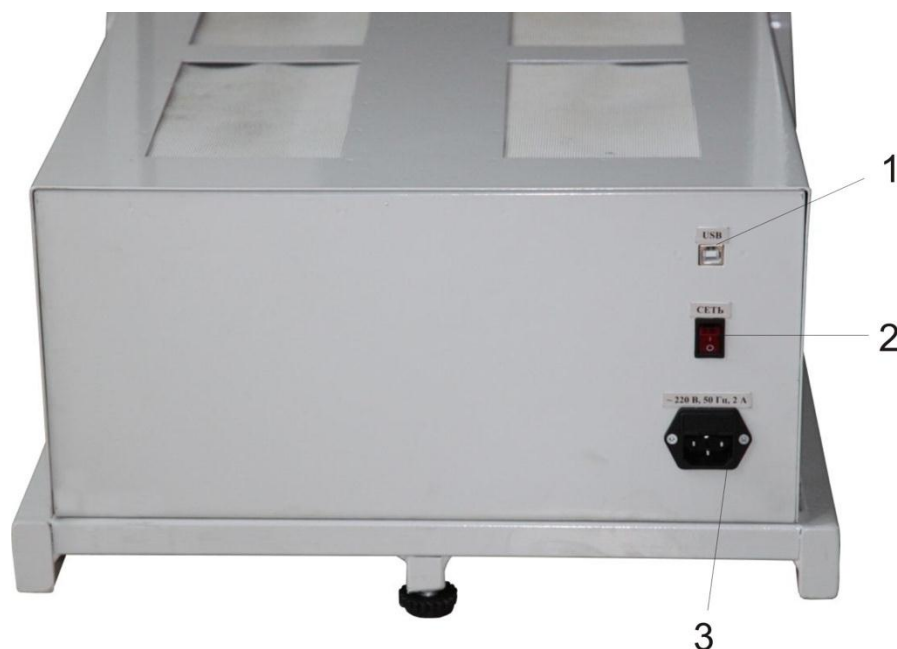


соот
ветс
твен
но.
Рис
уно
к
3.5
–
Лиц
евая
пан
ель
БП

На рисунке 3.5 номерами обозначены:

- 1- разъем подключения поворотного устройства;
- 2- разъемы 1-4, 5-8, 9-12, 13-16 подключение излучателей антенного блока ФАР;

- 3- разъемы 1-2 и 3-4 подключения приемников антенного блока;
- 4- индикаторы работы установки;
- 5- подключение мобильной антенной системы (приемника или излучателя на стойке);
- 6- разъем подключения рамы с поворотным подвесом.



- 1- разъем USB для связи с блоком визуализации и внешнего управления (ПЭВМ);
- 2- выключатель питания с подсветкой «Сеть»;
- 3- сетевой разъем для подключения к сети 220 В, 50 Гц с предохранителем.

Рисунок 3.6 – Задняя панель БППУ

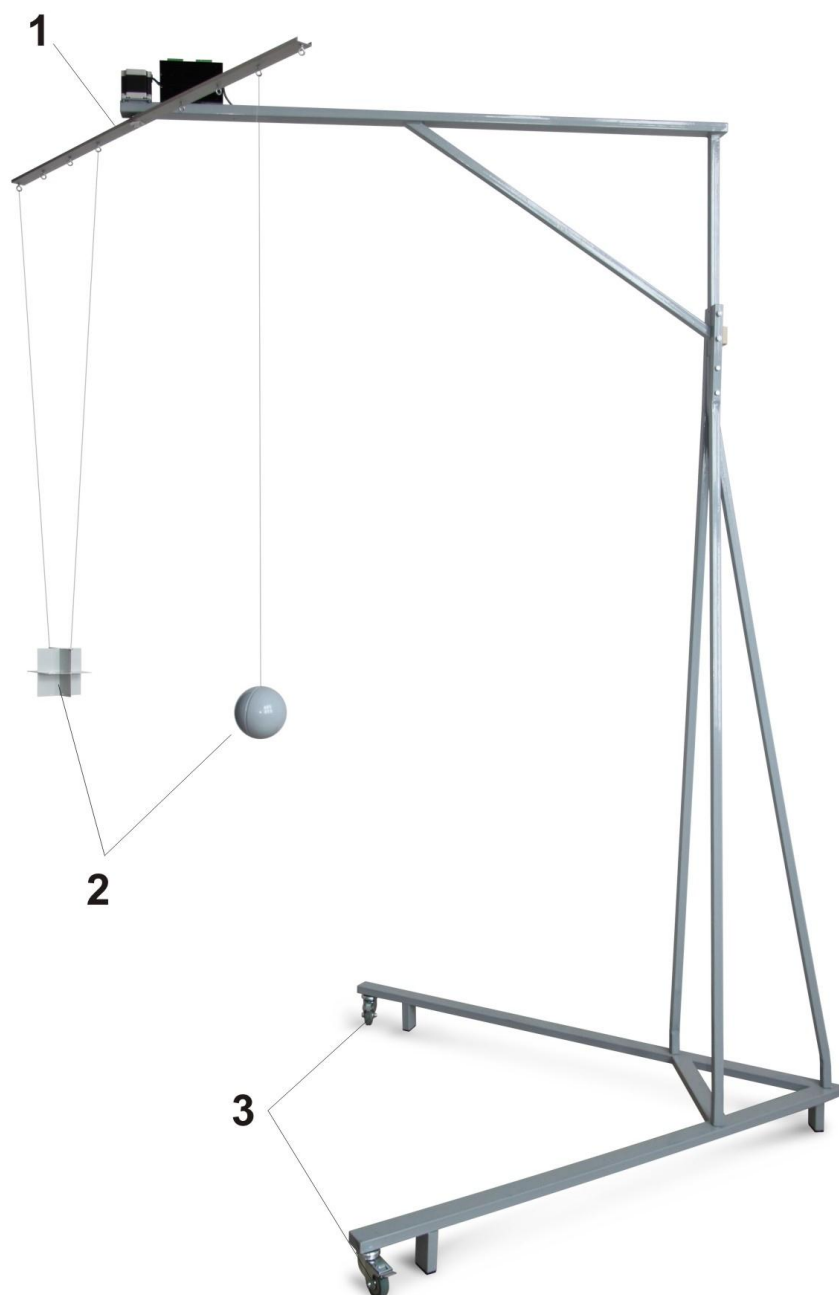
3.3 Рама с поворотным подвесом

Рама с поворотным подвесом и объектами обнаружения изображена на рисунке 3.7.

Характеристики:

- возможность перемещения рамы по полигону;
- поворот/вращение объекта обнаружения вокруг собственной оси на 360 градусов с шагом 1,8 градуса;

– поворот/вращение объекта обнаружения по окружности диаметром до 0,98 метра в диапазоне угловых скоростей от 0 до π радиан/с.



- 1- поворотный подвес;
- 2- объекты обнаружения;
- 3- колеса.

Рисунок 7 – Рама с поворотным подвесом и объектами обнаружения

Изм.	Лист	№ докум.	Подпись	Дата

3.4 Комплектация

Комплект учебно-лабораторного комплекса соответствует, указанному в таблице 3.1.

Таблица 3.1 – Комплектация

Наименование и условное обозначение изделия	Количество	Примечание
Радиолокационная стойка в сборе	1 шт.	-
Мобильная антенная система	1 шт.	Состав: микрофонная стойка, излучатель, приемник
Рама с поворотным подвесом для объектов обнаружения	1 шт.	Поставляется в разобранном виде
Блок визуализации и внешнего управления	1 шт.	ПК: процессор Intel Core i5 (частота 3,2 ГГц, в режиме Turbo 3,6 ГГц)/ Gigabyte GA-H110M-S2LGA / DDR4 4 Гб/ HDD 500 Гб/клавиатура USB/мышь USB/монитор Acer 21,5”(1920×1080)/ ОС Astra Linux
Комплект целей для обнаружения	1 комплект	1. Различающиеся по форме – шесть видов (пирамида, шар, лист, угловой отражатель двух видов, модель самолета). 2. Различающиеся по размерам (металлический лист) – 5 шт. 3. Различающиеся по материалам (метал, фанера, оргстекло) – 3 шт.
Сетевой кабель, 5 м	1 шт.	Для подключения питания к радиолокационной стойке
Сетевой фильтр	1 шт.	10 А, 6 розеток
Кабель USB Am-Bm, 3 м	1 шт.	Для подключения установки АФАР к ПК
Патч-корд UTP 5e, 1.5 м	6 шт.	Для подключения антенного блока, маркирован с двух сторон
Патч-корд UTP 5e с усилителем линии, 5 м	1 шт.	Для подключения мобильной антенной системы
Руководство по эксплуатации лабораторного комплекса	1 экз.	Согласно стандарту предприятия ОТСО-076.000 РЭ
Рекомендации по проведению экспериментов	1 экз.	

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ

Лист

33

4 Анализ функциональной схемы устройства

Из БППУ учебно-лабораторного комплекса можно выделить такие основные части, как: управляющий микроконтроллер; преобразователь интерфейса; передающую часть, включающую 16 цифровых синтезаторов, 16 цифровых резисторов, 16 линейных усилителей и 16 ультразвуковых передатчиков; приемную часть, включающую два ультразвуковых приемника, два логарифмических усилителя и два АЦП. Функциональную схему можно увидеть в документе ЮУрГУ–Д.11.05.01.2019.307.00 Э2.

Управляющим микроконтроллером является микроконтроллер фирмы STMicroelectronics STM32F407VGT6, предназначенный для решения вычислительных задач и передачи данных в ПК. Главным критерием выбора контроллера со схемотехнической точки зрения является достаточное количество выводов и интерфейсов, высокая производительность, компактность корпуса, а также удобство пайки и монтажа элемента в целом.

Внешний вид корпуса LQFP100 управляющего контроллера с указанием выводов представлен на рисунке 4.1:

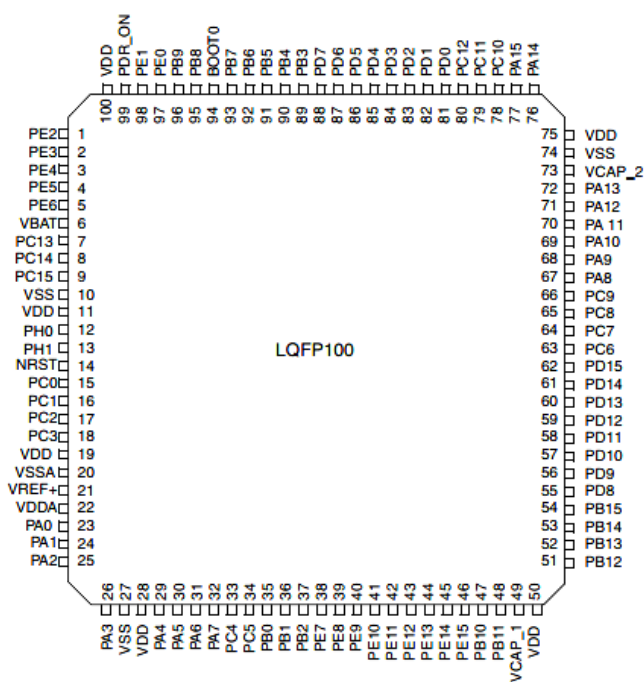


Рисунок 4.1 – Внешний вид микроконтроллера

Основные преимущества выбранного микроконтроллера:

- низкая стоимость семейства;
- наличие необходимых интерфейсов в достаточном количестве;
- компактный корпус;
- высокая тактовая частота и как следствие высокая производительность контроллера.

Характеристики выбранного микроконтроллера STM32F407VGT6:

- ядро: ARM Cortex-M4, 32-бит;
- flash-память: 1 Мбайт;
- RAM-память: 192 кБайт;
- аппаратное управление дисплеем: интерфейсы 6800/8080;
- кол-во каналов АЦП: 24 шт.;
- разрешение АЦП: 12 бит;
- скорость АЦП: 2,4 МГц;
- кол-во каналов ЦАП: 2 шт.;
- разрешение ЦАП: 12 бит;
- разрешение ШИМ: 16 бит;
- сторожевой таймер: 2 шт.;
- аппаратный модуль RTC: 1 шт.;
- кол-во интерфейсов UART: 6 шт.;
- кол-во интерфейсов CAN: 2 шт.;
- кол-во интерфейсов USB-хост: 2 шт.;
- кол-во интерфейсов USB-устройство: 2 шт.;
- кол-во интерфейсов ETHERNET: 1 шт.;
- напряжение питания периферии: 1,8...3,6 В;
- рабочий диапазон температур: от минус 40 до плюс 85 °С;
- чувствительность к влажности: уровень MSL 3 – 168 часов;
- вес: 2,243 г.

В качестве приемопередатчиков выбраны приемопередатчики фирмы Murata MA40S4S/R, которые имеют требуемую диаграмму направленности, с применением экранированных кабелей, внешняя оболочка которых заземлена для устранения помех.

Внешний вид приемопередатчиков схож, различия имеются лишь в их внутреннем строении. Графическое изображение передатчика и приемника представлено на рисунке 4.2:



Рисунок 4.2 – Внешний вид передатчика и приемников

Основные рабочие характеристики передатчика:

- минимальный диапазон срабатывания: 200 мм;
- максимальный диапазон срабатывания: 4000 мм;
- рабочая частота: 40 кГц;
- диаметр корпуса: 10 мм;
- диапазон температур: от минус 40 до плюс 85 °С.

Ультразвуковой передатчик имеет малую мощность излучения, соответствующую нормам безопасности. Экспериментально установлено, что диаграмма направленности передатчика и приемников составляет около 90 градусов.

Сигналы, передаваемые в передатчики, генерируются цифровыми синтезаторами. Частоту и фазу устанавливает микроконтроллер по шине SPI, также по шине SPI к каждому цифровому резистору передается требуемая амплитуда генерируемых сигналов. Перед тем, как передать сигнал в передатчик, он усиливается линейным усилителем.

5 Программный раздел

В данном разделе приводится обоснование применения функций микроконтроллера. Выбор языка программирования микропроцессорной и компьютерной программных сторон приведен в разделе анализа технического задания. Выделены основные этапы разработки и написания кода.

5.1 Разработка программы для микроконтроллера

Программа для микроконтроллера разрабатывается в программной среде Eclipse IDE, включающей дополнительный программный комплекс CDT (C/C++ Developing Tooling). Eclipse IDE является кроссплатформенной, в частности, работающей в ОС Linux. Выбор данной среды обусловлен ее бесплатным распространением и использованием, а также удобством интерфейса.

Основное окно программы разделено на несколько областей, каждое из которых несет определенный функционал. В левой части окна расположена область, содержащая структура проекта. Основной областью служит область ввода программного кода. Переключение между файлами при помощи вкладок позволяет быстро просмотреть, перенести или скопировать информацию между разрабатываемыми частями проекта. Язык программирования в данной среде – C++. Большое распространение языка делает его универсальным средством программирования и переноса кода между платформами большинства устройств [8], [9].

Пример интерфейса программы изображен на рисунке 5.1.

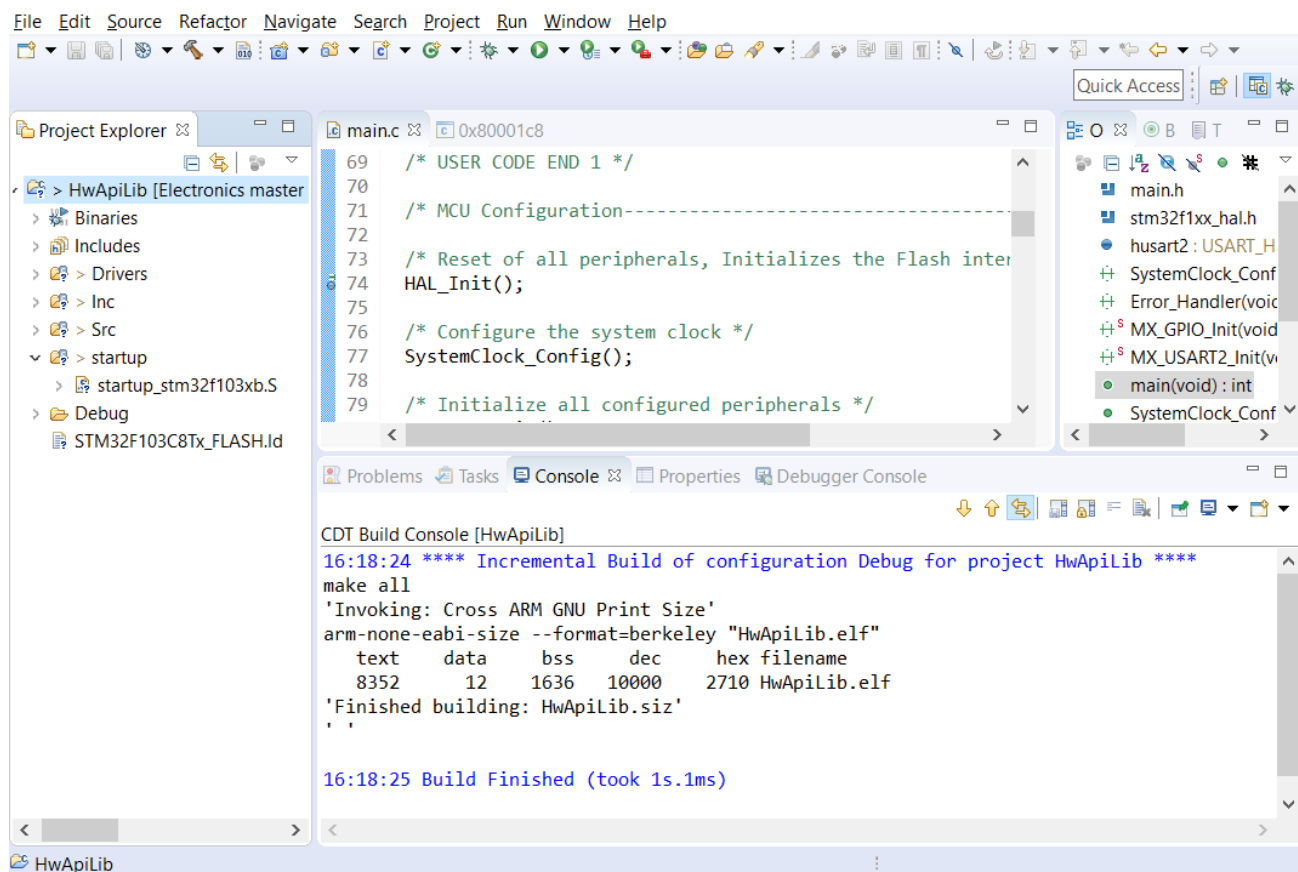


Рисунок 5.1 – Графический интерфейс программы Eclipse IDE

В среде программирования Eclipse IDE возможно использование большого количества подключаемых библиотек, которые содержат описание регистров выбранного в работе микроконтроллера. При их использовании отпадает необходимость объявления адресов используемых регистров памяти, код становится более читабельным и простым.

В данной работе используется одна подключаемая библиотека CMSIS контроллера STM32F407VGT6 которая представляет собой заголовочный файл "stm32f4xx.h". Использование других библиотек нецелесообразно, так как реализуемые в них функции не представляются оптимальными и не содержат необходимых алгоритмов расчета [10], [11], [12].

Приведенный в приложении данной дипломной работы код управляющего микроконтроллера является общим алгоритмом всех применяемых функций контроллера, которые поочередно выполняются в одном

бесконечном цикле. Для удобства весь процесс создания кода разделяется на несколько стадий:

- подключение необходимых файлов и библиотек;
- инициализация переменных;
- описание функций;
- основной цикл программы для вычислений.

В процессе создания программы составлен ее алгоритм и блок-схема основных взаимодействующих структурных частей, которые приведены на листах формата А1, прилагаемых к дипломной работе.

Также в процессе создания программы была разработана система команд общения микроконтроллера и персонального компьютера. Для всех используемых команд, микроконтроллер является ведомым, т.е. он только отвечает на запросы с ПК, данные передаются на ПК по запросу.

Рассмотрим на примере работу команды получения пачки из 40 импульсов длительностью 1 мс.

ПК отправляет микроконтроллеру команду $88 + DD$, где DD – номер приемника. Микроконтроллер принимает команду, далее проверяет, поддерживается ли эта команда, а также корректные ли данные присылаются. Если команда прошла проверку, то далее происходит установка частоты и фазы излучателей. Для этого по шине SPI происходит настройка каждого цифрового синтезатора, генерирующих аналоговые сигналы заданной частоты и с заданной фазой, а также по другой шине SPI происходит настройка цифровых резисторов, регулирующих амплитуду сигналов каждого излучателя. От цифровых резисторов сигналы передаются к линейным усилителям, а далее к ультразвуковым излучателям. Сигналы излучаются в диапазоне частот 39...41 кГц.

Далее отраженный от цели сигнал принимается одним или двумя ультразвуковыми приемниками (в зависимости от заданного режима работы).

					ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		39

Сигналы проходят через логарифмические усилители, позволяющие выровнять амплитуды. Затем сигналы оцифровываются и передаются в микроконтроллер. Оцифрованные сигналы микроконтроллер записывает в буфер для последующей передачи в ПК.

5.2 Разработка программы для компьютера

Программа управления и визуализации данных разрабатывается на персональном компьютере в программной среде IntelliJ IDEA на языке программирования Kotlin. Обоснование выбора этой среды и языка программирования приведены в раздел анализа технического задания.

Программа предназначена для:

- а. управления поворотным подвесом:
 1. выбор направления поворота/вращения;
 2. однократный поворот на некоторый угол с шагом 1.8° ;
 3. непрерывное вращение с угловой скоростью от 0 до 1 об/с с шагом 1/100;
 4. остановка вращения цели;
- б. выбора режима работы РЛС:
 1. импульсная амплитудная РЛС (1 луч);
 2. импульсная амплитудная РЛС (2 луча);
 3. РЛС с линейной частотной модуляцией (ЛЧМ) и непрерывным излучением;
 4. импульсная фазовая РЛС;
- в. задания алгоритма работы в соответствии с выбранным режимом работы;
- г. ручного управления АФАР:
 1. включение/выключение отдельных излучателей;
 2. задание начальных фаз излучения;
 3. задания амплитуд излучателей;

									Лист
									40
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ				

д. Приема, обработки данных с одного или двух приемников (в зависимости от режима работы);

е. Графического отображения принятого сигнала и результатов обработки.

Разрабатываемая программа имеет одно графическое окно, которое отображается после запуска программы. Окно разделено по вертикали на две области, в каждой из которых присутствуют вкладки. В левой области присутствуют вкладки для управления целью и осциллографом. В правой – вкладки отображения осциллографа, формирование луча и обработка принятого сигнала.

Внешний вид программы с активными вкладками управления целью и осциллограммой приведен на рисунке 5.1.

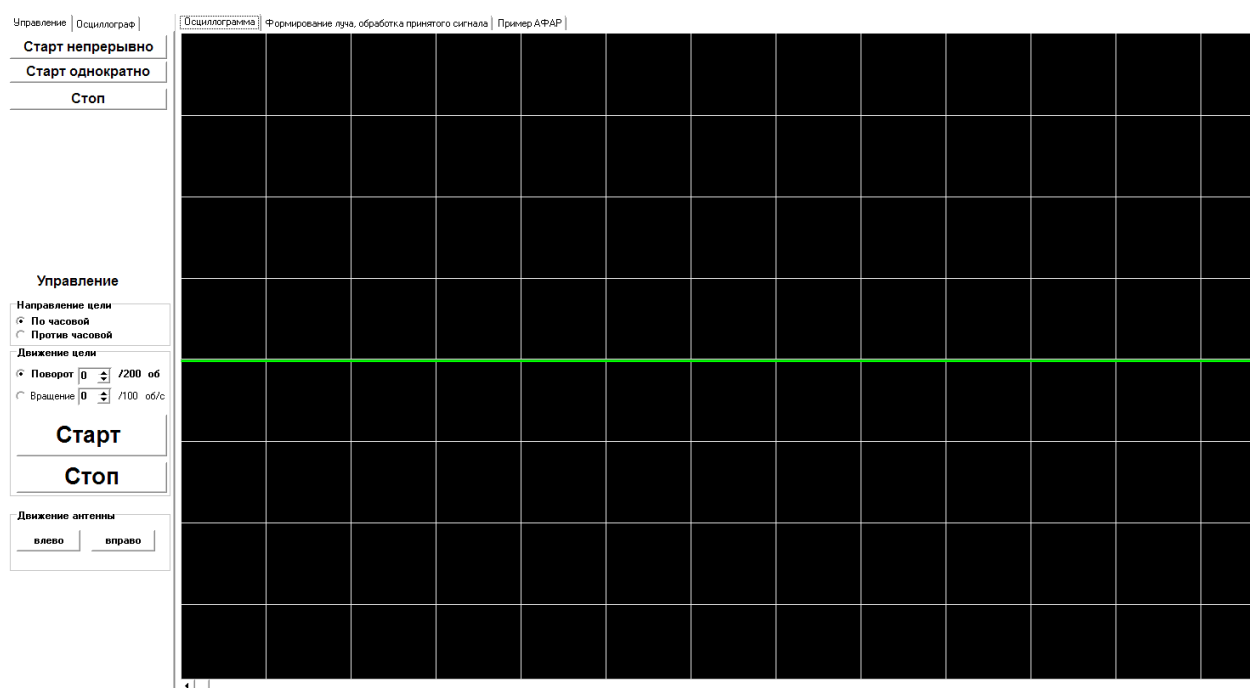


Рисунок 5.1 – Внешний вид программы при запуске

В левой области на вкладке «Управление» находятся следующие кнопки.

«Старт однократно» – выполняет единичный запрос к БППУ в соответствии с выбранным алгоритмом работы. При этом происходит следующее:

- процессор БППУ формирует сигналы управления для передающей ФАР;
- процессор БППУ начинает оцифровку принятого сигнала с момента начала передачи;
- по окончании оцифровки процессор передает данные в компьютер для анализа.

«Старт непрерывно» – запускает непрерывный цикл «однократных запросов» к БППУ с частотой около 15 Гц.

«Стоп» – останавливает работу РЛС (передатчиков и приемников).

Эти же кнопки продублированы на вкладке осциллографа.

Ниже расположены элементы для управления подвесом цели, с помощью которых можно выбрать направление движения (по часовой против часовой), задать поворот или непрерывное вращение цели.

Поворот цели можно выбрать из диапазона от 0 до 200 единиц. Единица соответствует повороту на 1,8 градуса. Полный оборот $2\pi = 200$.

Вращение цели можно выбрать из диапазона от 0 до 100. Получаемая угловая скорость может быть определена, как $\omega = 2 \cdot \pi \cdot X/100$, где X – установленная в соответствующем поле скорость.

Кнопка «Старт» – запускает подвес цели по выбранному алгоритму (поворот или вращение).

Кнопка «Стоп» – останавливает вращение.

«Движение антенны» имеет две кнопки «вправо» и «влево». Нажатие на кнопку приводит к повороту антенны по/против часовой стрелки на заданное количество шагов. Минимальный шаг поворота соответствует повороту на $\pi/1600$.

На рисунке 5.2 изображена программа с активной вкладкой «Осциллограф» и осциллограммой, полученной приемником.

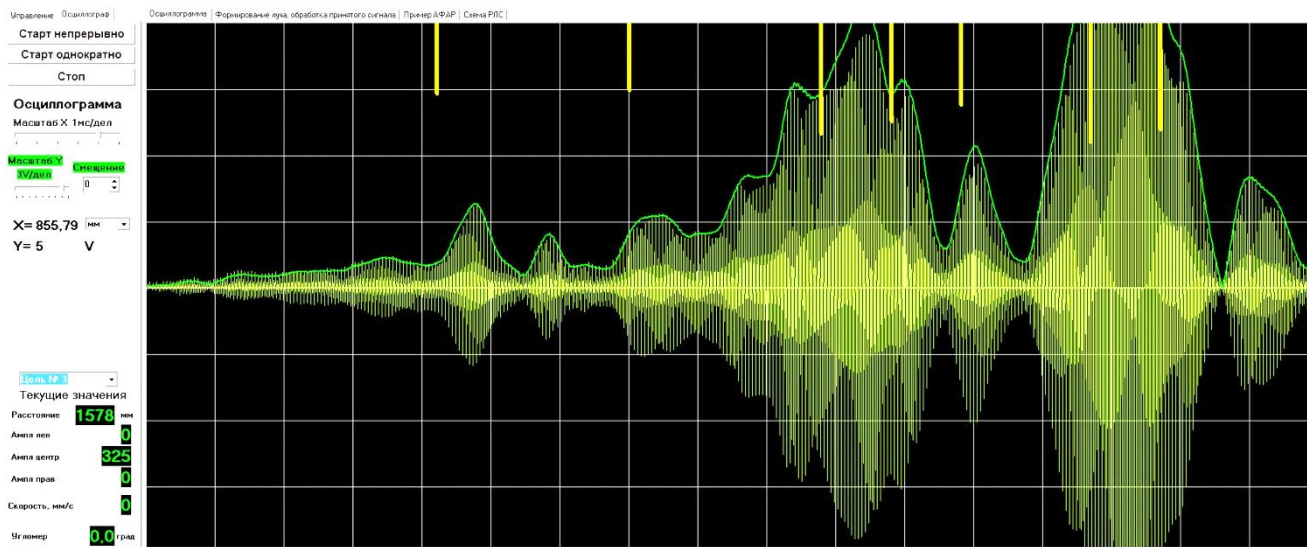


Рисунок 5.2 – Экран осциллограммы в работе

На осциллограмме сам сигнал нарисован зеленым цветом. Светло-зеленым цветом отображается огибающая амплитуды корреляции синусоидой 40 кГц. Красным, при необходимости, отображается фаза корреляции с образцовой синусоидой. Вверху желтым цветом отмечаются метки целей при текущих настройках.

Вкладку управления осциллографом можно рассмотреть более подробно

на рисунке 5.3.

Рисунок 5.3 – Вкладка управления осциллографом

Кнопки «Старт непрерывно», «Старт однократно», «Стоп» – выполняют те же функции, что и на вкладке управления целью.

«Масштаб X» – позволяет изменять развертку осциллографа.

«Масштаб Y» – позволяет изменять масштаб сигнала.

«Х» – показывает положение курсора мыши на оси Х осциллограммы в разных величинах – отсчетах, микросекундах или миллиметрах. (на рисунке 5.2 в миллиметрах).

«Y» – положение курсора мыши с учетом выбранного масштаба по Y.

«Движение антенны» повторяют функции движение антенны на вкладке управления целью.

«Цели» – динамический список обнаруженных в текущем замере при текущих настройках целей.

«Текущие значения» – относятся к выбранной текущей цели. Изменяются только те значения, которые в данный момент вычисляются:

«Расстояние» – в миллиметрах.

«Ампл лев» – амплитуда левого луча.

«Ампл центр» – амплитуда нормали к плоскости излучения.

«Ампл прав» – амплитуда правого луча.

«Ампл лев», «Ампл прав» используются при двухлучевой радиолокации. Определяется ближайший обнаруженный объект, и вычисляется максимальная амплитуда отраженного от него сигнала.

«Ампл центр» используется в однолучевой РЛС.

«Скорость» – вычисляется по доплеровскому сдвигу частоты.

Положение антенны – показывает угол поворота антенны относительно нулевого положения в градусах. Фактически угол поворота измеряется с точностью градуса ($\pi/1600$).

Содержимое вкладки «Управление излучателями» изображено на рисунке 5.4.

Управление излучателями

ВКЛ	ФАЗА	Амплитуда
<input type="checkbox"/> 1	0	255
<input type="checkbox"/> 2	0	255
<input type="checkbox"/> 3	214	255
<input type="checkbox"/> 4	143	255
<input type="checkbox"/> 5	71	255
<input type="checkbox"/> 6	0	255
<input type="checkbox"/> 7	0	255
<input checked="" type="checkbox"/> 8	0	255
<input checked="" type="checkbox"/> 9	0	255
<input type="checkbox"/> 10	0	255
<input type="checkbox"/> 11	0	255
<input type="checkbox"/> 12	0	255
<input type="checkbox"/> 13	0	255
<input type="checkbox"/> 14	0	255
<input type="checkbox"/> 15	0	255
<input type="checkbox"/> 16	0	255
<input type="checkbox"/> ON		
<input type="checkbox"/> OFF	Сброс	Сброс

Рисунок 5.4 – Вкладка управления излучателями

Кнопки «ON» и «OFF», позволяют включить или выключить все излучатели одновременно.

Секция «Фаза» – позволяет задать фазу излучения каждого излучателя в отдельности. С помощью начальных фаз формируется электронное отклонение луча от нормали к плоскости излучения. Фаза может принимать значения от 0 до 4095 и может быть вычислена по формуле:

$$\varphi = 2 \cdot \pi \cdot X / 4096,$$

где X – значение, установленное в поле «фаза».

Кнопка «Сброс» в секции «Фаза» устанавливает все фазы в нулевое состояние.

Секция «Амплитуда» позволяет установить выходную амплитуду каждого излучателя в отдельности. Допустимые значения от 0 до 255 (8 разрядов)

Кнопка «Сброс» в секции «Амплитуда» устанавливает все амплитуды в максимум (равномерное распределение).

На вкладке управления излучателями также имеется секция выбор режима работы РЛС, она изображена на рисунке 5.5 и секция настройки порога обнаружения, которая изображена на рисунке 5.6.

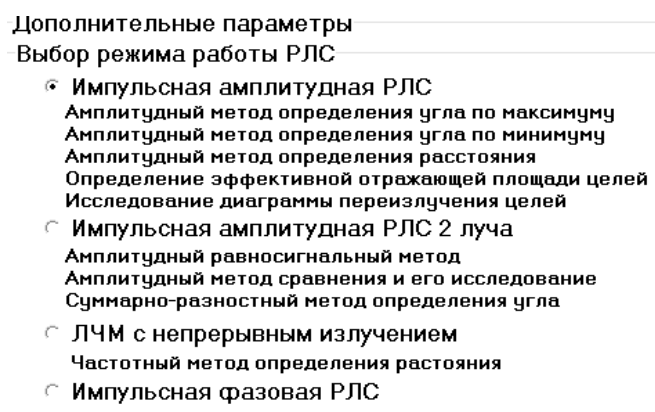


Рисунок 5.5 – Режим выбора РЛС

Выбор режима РЛС позволяет установить режим работы в соответствии с выполняемой лабораторной работой.

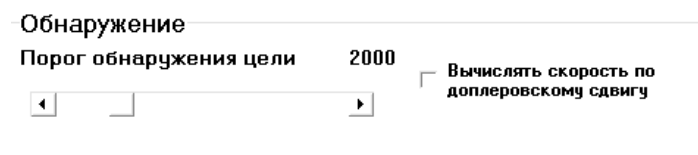


Рисунок 5.6 – Обнаружение

«Обнаружение» – позволяет изменять порог обнаружения цели. При увеличении порога слабо отражающие объекты не будут отмечаться желтой меткой цели. При увеличении порога – наоборот.

«Вычислять скорость по доплеровскому сдвигу» – позволяет включить или отключить вычисление скорости объекта. Вычисление скорости ресурсоемко и по умолчанию отключено.

6 Организационно-экономический раздел

В данном разделе приводится экономический анализ и сетевое планирование процесса разработки программного обеспечения. Оно заключается в выявлении оценки экономического эффекта от разработки обновленного программного обеспечения. В данном разделе выделены и рассчитаны основные этапы экономического планирования, построен сетевой график, наглядно отражающий время выполнения работ.

В конце раздела подведены итоги экономического анализа, в которых отображена эффективность разработки.

6.1 Понятие параметров сетевого планирования

Сетевой анализ является методом анализа сроков выполнения частей проекта, который позволяет связать выполнение различных задач и процессов во времени, получив прогноз продолжительности реализации проекта в целом. С его помощью выделяют наиболее рациональные пути развития этапов производства, от идеи до готовой единицы продукции, и получают объективную оценку этих параметров при выбранном варианте структуры работ и распределения ресурсов [13], [14].

Существуют различные методы построения сетевого планирования:

- а. детерминированные сетевые методы:
 1. диаграмма Ганта с дополнительным временным люфтом 10...20 %;
 2. метод критического пути (МКП);
- б. вероятностные сетевые методы:
 1. неальтернативные:
 - а. метод статистических испытаний (метод Монте-Карло);
 - б. метод оценки и пересмотра планов (PERT);
 2. альтернативные:
 - а. метод графической оценки и анализа (GERT).

Сетевое планирование и управление содержит три основных этапа: структурное планирование, календарное планирование и оперативное управление. Оно начинается с разбиения текущего проекта на определенные шаги, которые необходимы для достижения конечной цели разработки. Затем определяются оценки продолжительности работ и строится сетевой график, позволяющий, прежде всего, оценить временные характеристики проекта и входящих в него работ.

Сетевой график состоит из элементов двух видов – работ и событий, и позволяет наглядно представить временную структуру проекта. Он отображает взаимосвязи между работами внутри проекта и порядок их выполнения. График является направленным графом, в котором каждая работа представляется ориентированной дугой, а каждое событие – узлом. Событие в свою очередь определяется как момент времени, когда завершается одна работа и начинается другая.

При построении сетевого графика, в процессе разработки, учитываются основные правила:

- перед построением графика в технологической последовательности формируются правила выполнения начала и окончания работ;
- сетевой график строится от начала к окончанию при прямом расчете, и в обратном направлении при обратном расчете;
- в сетевом графике отсутствуют "тупиковые" события;
- в сетевом графике отсутствуют замкнутые контуры – пути, соединяющие события, связанные сами с собой.

Особое значение в сетевом графике имеют критические работы. Критической работой называют работу, задержка начала которой приводит к задержке конечного срока проекта в целом. Некритической работой – ту работу, где промежуток времени между ее ранним началом и поздним окончанием больше ее фактической продолжительности. То есть, любая не критическая

работа имеет резерв времени, что позволяет максимально оптимизировать её выполнение. Поэтому в данной дипломном проекте в экономических расчетах используется метод критического пути. Его преимуществом является то, что длительность операций можно оценить с достаточно высокой степенью точности и определенности. Основным достоинством метода критического пути является возможность манипулирования сроками выполнения задач, не лежащих на критическом пути.

Задачи, которые лежат на критическом пути, имеют нулевой резерв времени выполнения и в случае изменения их длительности изменяются сроки всего проекта. В связи с этим при разработке критические задачи требуют более тщательного контроля, в частности, своевременного выявления проблем и рисков, влияющих на сроки их выполнения и, следовательно, на сроки выполнения проекта в целом.

Календарное планирование по выбранному МКП требует определенных входных данных. После их ввода производится процедура прямого и обратного прохода по сети и вычисляется выходная информация.

6.2 Анализ этапов разработки, построение сетевого графика

Перед построением модели сетевого графика рассчитывают временные параметры всех составляющих его компонентов.

Для полного построения следует произвести два вида расчетов: прямой и обратный. При прямом расчете вычисляют ранние сроки выполнения каждого события. При обратном вычислении их производят в обратной последовательности, что позволяет вычислить поздние сроки событий. Также на сетевом графике отмечают номера событий и разницы во времени между ранними и поздними сроками.

Прямой расчет – определение минимально возможного времени реализации проекта начинается с работ, не имеющих предшественников. В ходе него определяется *ES* (ранний старт) и *EF* (ранний финиш). Ранние начала и

									Лист
									50
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ				

ранние окончания работ определяются последовательно, слева направо по графику, то есть от исходного события сети к завершающему.

Основными формулами для расчетов являются:

$$ES_0 = 0, \quad (6.1)$$

$$EF = ES + CONT, \quad (6.2)$$

где ES_0 – нулевой ранний старт,

ES – ранний старт события,

EF – ранний финиш события,

$CONT$ – продолжительность события.

Ожидаемое время выполнения работы ($t_{\text{ожид}}$) рассчитывается по двухоценочной методике, исходя из минимальной ($t_{\text{мин}}$) и максимальной ($t_{\text{макс}}$) оценок продолжительности работы. При этом предполагается, что минимальная оценка соответствует наиболее благоприятным, а максимальная – наиболее неблагоприятным условиям работы.

Ниже приведены основные формулы прямого расчета. Ожидаемая продолжительность каждой работы рассчитывается по формуле:

$$t_{\text{ожид}} = 0,6 \cdot t_{\text{мин}} + 0,4 \cdot t_{\text{макс}}. \quad (6.3)$$

Среднеквадратическое отклонение (δ_{ij}) продолжительности выполнения работ в двухоценочной методике определяется по формуле:

$$\delta_{ij} = 0,2 \cdot (t_{ij\text{мин}} - t_{ij\text{макс}}). \quad (6.4)$$

Дисперсия определяется по формуле:

$$D_{ij} = 0,04 \cdot (t_{ij\text{мин}} - t_{ij\text{макс}})^2. \quad (6.5)$$

Если при прямом расчете возникают ситуации выполнения двух или более параллельных процессов, то ранний срок свершения следующего промежуточного события вычисляется по функции максимума:

$$t_p = \max(t_{pi} + t_{ij}), \quad (6.6)$$

где t_{ij} – ожидаемая продолжительность работы,

t_{pi} – ранний срок свершения события, непосредственно предшествующего данному.

Обратный расчет. Определяются LS (поздний старт), LF (поздний финиш) и R (резерв). Поздние начала и поздние окончания определяются в обратном порядке – от завершающегося события графика к исходящему, то есть справа налево.

Основными формулами для расчетов являются:

$$EF_N = LF_N, \quad (6.7)$$

$$LS_i = LF_i - CONT, \quad (6.8)$$

где LF_N – поздний финиш конечного события,

LS_i – поздний старт события,

LF_i – поздний финиш события,

$CONT$ – продолжительность события.

После прохождения по сетевому графику в прямом и обратном направлениях должно выполняться условие $ES_0 = LS_0$. Это означает, что для завершающегося события никакие резервы времени не планируются. Если при обратном расчете возникают ситуации выполнения двух или более параллельных процессов, то поздний срок свершения следующего промежуточного события вычисляется по функции минимума:

$$t_s = \max(t_{si} + t_{ij}), \quad (6.9)$$

где t_{si} – ожидаемая продолжительность работы,

t_{ij} – поздний срок свершения события непосредственно следующего за данным промежуточным событием.

Резерв времени свершения события – это промежуток времени, на который может быть отсрочено событие без нарушения сроков разработки в

целом. Он образуется у событий, для которых поздний срок свершения события больше раннего срока, то есть имеется свободный временной промежуток:

$$R_i = t_s - t_p. \quad (6.10)$$

Если $t_s = t_p$, то есть поздний срок свершения события равен раннему сроку свершения события, то такое событие не имеет резерва времени и это событие относится к критическому пути.

Результаты расчетов временных параметров событий сетевого графика приведены в таблице 6.1.

Таблица 6.1 – Проведение работ по разработке стенда

Код работы	Наименование работы	Продолжительность, дн.			Исполнители, чел.			δ_{ij} , дн.	D_{ij} , дн ²
		мин.	макс.	ожид.	руков.	инж.	инж.		
0,1	Получение технического задания на разработку ПО	1	2	1	1	1	1	0,20	0,04
1,2	Выбор литературных источников	3	4	3	0	1	1	0,20	0,04
1,4	Определение стадий и этапов разработки ПО	5	7	6	1	0	0	0,40	0,16
2,3	Изучение структурной схемы цифровой части стенда	1	2	1	0	1	1	0,20	0,04
3,4	Изучение принципиальной схемы	1	2	1	0	1	1	0,20	0,04
4,5	Разработка протокола общения устройства с компьютером	4	5	4	0	1	1	0,20	0,04
5,6	Определение средств разработки программы для ПК	3	5	4	1	1	0	0,40	0,16
5,7	Создание программы для управляющего микроконтроллера	10	14	12	0	0	1	0,80	0,64
6,7	Создание программы для персонального компьютера	8	12	10	0	1	0	0,80	0,64
7,8	Отладка программы микроконтроллера	3	6	4	0	1	1	0,60	0,36
8,9	Отладка программы персонального компьютера	1	2	1	0	1	1	0,20	0,04

Продолжение таблицы 6.1

Код работы	Наименование работы	Продолжительность, дн.			Исполнители, чел.			δ_{ij} , дн.	D_{ij} , дн ²
		мин.	макс.	ожд.	руков.	инж.	инж.		
7,9	Выявление достоинств и недостатков алгоритмов, оптимизация их работы	5	7	6	1	1	1	0,40	0,16
9,10	Оформление программной документации	2	3	2	1	1	1	0,20	0,04
10,11	Анализ литературы организационно-экономического раздела	2	4	3	0	1	0	0,40	0,16
11,12	Разработка и анализ сетевого графика	2	3	2	1	1	0	0,20	0,04
12,13	Расчет прибыли и экономического эффекта разработки	3	5	4	1	0	0	0,40	0,16
13,16	Оформление организационно-экономического раздела	4	6	5	0	1	0	0,40	0,16
10,14	Анализ литературы раздела БЖД	2	3	2	0	0	1	0,20	0,04
14,15	Разработка мер безопасности человека от негативных факторов при работе со стендом	2	5	3	0	0	1	0,60	0,36
15,16	Оформление раздела БЖД	3	4	3	0	0	1	0,20	0,04
16,17	Оформление технической документации	3	5	4	1	1	1	0,40	0,16
16,18	Оформление пояснительной записки	6	10	8	0	1	1	0,80	0,64
17,18	Оформление графических документов дипломного проекта	4	6	5	1	1	1	0,40	0,16
18,19	Проверка и сдача проекта	2	5	3	1	1	1	0,60	0,36

После построения сетевого графика проводится его всесторонний анализ, с тем чтобы в дальнейшем принять меры по его оптимизации. Определить степень трудности выполнения в срок каждой группы работ не критического пути можно с помощью коэффициента напряженности работ. Коэффициентом

напряженности K_H работы называется отношение продолжительности несовпадающих (заключенных между одними и теми же событиями) отрезков пути, одним из которых является путь максимальной продолжительности, проходящий через данную работу, а другим – критический путь:

$$K_H = \frac{t(L_{\max}) - t'_{кр}}{t_{кр} - t'_{кр}}, \quad (6.11)$$

где $t(L_{\max})$ – продолжительность максимального пути, проходящего через работу;

$t_{кр}$ – продолжительность отрезка рассматриваемого пути, совпадающего с критическим путем;

$t'_{кр}$ – продолжительность критического пути.

Коэффициент сложности сетевого графика представляет собой отношение количества работ сетевого графика к количеству событий и определяется по формуле:

$$K_{\text{сложн}} = n_{\text{раб}} / n_{\text{соб}}, \quad (6.12)$$

где $K_{\text{сложн}}$ – коэффициент сложности сетевого графика;

$n_{\text{раб}}$ – количество работ, ед.;

$n_{\text{соб}}$ – количество событий, ед.

Сетевые графики, имеющие коэффициент сложности от 1,0 до 1,5, являются простыми, от 1,51 до 2,0 – средней сложности, более 2,1 – сложными.

Для данного графика коэффициент сложности не превышает 1,5, график является простым:

$$K_{\text{сложн}} = 24/20 = 1,2. \quad (6.13)$$

Согласно приведенной выше таблице и расчетным данным, приводится сетевой график с рассчитанными значениями затрат по времени на каждую стадию разработки, он приведен на отдельном листе А1, который прилагается к пояснительной записке.

6.3 Расчет разработки ПО

Амортизация. Она выражается в износе оборудования для производства.

Срок полезного использования персонального компьютера, на котором осуществлялось программирование составляет 5 лет. Стоимость компьютера составляет 30000 рублей вместе с операционной системой «Windows 10 Домашняя». Эксплуатация компьютера длилась 2,5 месяца. Согласно приведенным данным получаем:

$$P_{\text{использ}} = \frac{T_{\text{использ}}}{T_{\text{общ}}} = \frac{2,5 \cdot 30000}{59} = 1271 \text{ рублей.}$$

Ежемесячная амортизация равна:

$$P_{\text{ежемес}} = \frac{30000}{59} = 508 \text{ рублей.}$$

При расчете амортизации учитывается, что разработка программного обеспечения осуществляется на территории самого предприятия, поэтому в амортизационные расходы нет необходимости включать стоимость арендной платы помещения.

Затраты на оплату труда включают в себя прямую заработную плату рабочим, премии, а также надбавку заработной платы в виде уральского регионального коэффициента. Итоговая заработная плата персонала будет рассчитываться по формуле:

$$ЗП_{\text{полн}} = (ОК + ЗП_{\text{прем}} + ЗП_{\text{коэф}}) \cdot t,$$

где $ЗП_{\text{полн}}$ – полная заработная плата, руб.;

$ОК$ – оклад, руб./день;

$ЗП_{\text{прем}}$ – премия, руб./день;

$ЗП_{\text{коэф}}$ – региональный уральский коэффициент, руб./день;

t – трудоемкость работ, дней.

Премияльная ставка равна 15% от прямой заработной платы работника. Уральский региональный коэффициент работ также составляет 15%. Исходя из этих данных приводится расчет полной заработной платы в таблице 6.2.

Таблица 6.2 – Трудоемкость и заработная плата исполнителей работ

Исполнители	Оклад, руб./мес.	Премия, руб.	Трудоемкость, дн.	Уральский региональный коэффициент (15%), руб.	Полная заработная плата, руб
Руководитель	35000	5250	20	5250	41364
Инженер	25000	3750	39	3750	57614
Инженер	25000	3750	38	3750	56136
Итого					115114

К прочим затратам относятся незначительные в рамках разработки проекта расходы, которые возникают на определенных этапах реализации продукции. К ним можно отнести расходы на печать документации и отчетность. Исходя из стоимости печати одного чертежа формата А1 20 руб. за лист, и одного листа формата А4 2 руб. за лист, получаем:

$$P_{\text{расх.мат}} = 5 \cdot 20 + 120 \cdot 2 = 340 \text{ рублей.}$$

К прочим затратам также относится энергопотребление ПК. В процессе программирования необходимо обеспечение питания компьютера от сети 220 Вольт. При расчете использовались данные по городу Челябинску, стоимость электроэнергии составляет 3,14 руб/кВт·час. Системный блок ПК потребляет 0,4 киловатта. Соответственно, затраты на электроэнергию:

$$P_{\text{эл.}} = T \cdot 3,14 \cdot P_{\text{потреб}} = 776 \cdot 3,14 \cdot 0,4 = 975 \text{ рублей,}$$

где $P_{\text{эл.}}$ – затраты на электроэнергию,

T – время работы ПК в часах,

$P_{\text{потреб}}$ – потребление ПК в кВт.

Итоговая стоимость разработки программного обеспечения на предприятии составляет

$$C = 1271 \cdot 2,5 + 115114 + 340 + 975 = 119607 \text{ рублей.}$$

6.4 Расчет экономического эффекта

Экономический эффект характеризует стоимостную оценку результативности выполненной работы.

Исходя из технического задания, новое разрабатываемое программное обеспечение для компьютера будет работать на операционной системе Astra Linux, которая распространяется бесплатно. Предыдущая версия ПО работает на операционной системе Microsoft Windows, которая является коммерческой. Стоимость продукта «Windows 10 Домашняя» составляет 9499 рублей [15].

Также стоит учесть, что разработка нового программного обеспечения велась на компьютере с ОС Linux, в этом случае стоимость разработки уменьшится на стоимость продукта Windows 10. Полная стоимость разработки будет составлять 110108 рублей. Эта сумма и будет экономическим эффектом проекта, описанного в данном дипломном проекте.

						Лист
					ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	58
Изм.	Лист	№ докум.	Подпись	Дата		

7 Безопасность жизнедеятельности

7.1 Анализ достоинств интерфейса пользователя разработанной программы

Пользовательский интерфейс для каждого пользователя отображается одинаково вне зависимости от настроек и не меняется в процессе работы с программой, поэтому любому из них будет легко адаптироваться пользоваться разработанным продуктом.

Для минимизации ввода информации, а также для предотвращения сбоев программы, вследствие ввода неконтролируемой величины, используются раскрывающиеся списки с вариантами выбора, поля ввода чисел с ограничениями минимального и максимального значений.

В продукте предусмотрены подсказки для кнопок, полей ввода и иных элементов, назначение которых интуитивно непонятно пользователю.

Предусмотрена блокировка действий пользователя, в виде отключенные элементов пользовательского интерфейса. Для каждого подобного случая присутствует пояснение в виде сообщений.

Любой сбой, возникший в разработанной программе или аппаратной части, имеет пояснение.

7.2 Рекомендации по организации рабочего места пользователя

Негативное влияние на человека ПК менее выражено, сглажено многочисленными положительными факторами. Но у людей, длительно использующих ПК, могут быть выявлены такие реакции как нарушение функций зрения, быстрое общее утомление. Для того чтобы избежать вредного воздействия при работе с вычислительной техникой необходимо соблюдать соответствующие меры безопасности, правильно планировать рабочее место и режим работы.

						Лист
					ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	59
Изм.	Лист	№ докум.	Подпись	Дата		

ПК может являться источником целого ряда вредных и опасных факторов производства: электромагнитных полей, статического электричества. Часто условия труда при работе на ПК усугубляются высокими уровнями шума, плохими микроклиматическими условиями и недостаточной освещенностью на фоне зрительного и нервно-эмоционального напряжения.

Работа на ПК может сопровождаться малой двигательной активностью.

Условия труда пользователя, который работает с ПК, определяются:

– отличительными чертами основных элементов рабочего места (пространственные параметры рабочего места и его элементов, соответствующие анатомии-физиологическим данным пользователя; расположение элементов рабочего места относительно пользователя с учетом типом работы);

– условиями окружающей среды (освещение в помещении зала с дисплеями и на рабочем месте, микроклимат, шум, другие факторы, обусловленные особенностями средств отображения информации, и т.д.);

– характеристиками информационного взаимодействия пользователя и ПК.

Усложнение функциональной структуры деятельности в связи с применением электронно-вычислительных систем, персональных компьютеров предъявляет новые, повышенные требования к организму человека. Недооценка значимости человеческого фактора при проектировании и создании ПК неизбежно отражается на качественных и количественных показателях деятельности пользователей:

– процедуры по вводу данных характеризуются значительной скоростью переработки информации, значительным темпом работы, невысокой потребностью в обмене информацией и незначительной частотой принятия решений. Работа не требует высокого умственного и зрительного напряжения, но сопровождается локальными мышечными нагрузками;

– диалоговые виды работ (редактирование, правка, верстка и др.) характеризуются средней скоростью обработки информации, неритмичной

									Лист
									60
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ				

потребностью в обмене информацией с ПК, которая сопровождается принятием решений.

Характерной чертой работы на ПК является постоянное и высокое напряжение функций зрения, которое обусловлено необходимостью выявления объектов (символов, знаков и т.п.), при наличии на экране: строчной структуры экрана, мелькания изображений, недоосвещенности поля монитора, недостаточной контрастности объектов и необходимости постоянной переадаптации зрения к различным уровням освещенности монитора, документа и клавиатуры.

Нервно - эмоциональное напряжение при работе на ПК возникает вследствие недостатка времени, значительного объема и плотности информации, особенностей диалогового режима общения человека и ПК (сбои, оперативное ожидание, психологические особенности работы пользователя, связанные с эмоционально - волевой сферой), ответственности за безошибочность информации.

Скорость работы на ПК при вводе информации (текста, данных и т.п.) определяется объемом и характером задания, и временем его выполнения.

В промежуток выполнения операций ввода данных количество мелких шаблонных движений кистей и пальцев рук за смену может превысить 60 тыс., что в соответствии с гигиенической классификацией труда относится к категории вредных и опасных.

7.2.1 Требования к производственным зданиям и помещениям

Производственные здания и помещения должны удовлетворять требованиям СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы», раздел «Требования к помещениям для работы с ПЭВМ».

						Лист
					ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	61
Изм.	Лист	№ докум.	Подпись	Дата		

В помещениях без естественного освещения эксплуатация ПК допускается только при наличии расчетов, которые обосновывают соответствие нормам естественного освещения.

Естественное и искусственное освещение должно соответствовать требованиям, описанным в подразделе «Требования к естественному и искусственному освещению» данной работы. Окна в помещениях, где эксплуатируется вычислительная техника, в большей степени должны быть ориентированы на север и северо-восток.

Оконные проемы должны быть оборудованы регулируемыми устройствами вида: жалюзи, занавесы, внешних козырьков и др.

Для внутренней отделки интерьера помещений, где расположены ПК, должны использоваться рассеивающие материалы с коэффициентом отражения для потолка – 0,7...0,8; для стен – 0,5...0,6; для пола – 0,3...0,5.

Полимерные материалы используются для внутренней отделки интерьера помещений с ПК при наличии санитарно-эпидемиологического заключения.

Помещения, в которых размещаются рабочие места с ПК, должны быть оборудованы защитным занулением в соответствии с техническими требованиями по эксплуатации.

Не следует размещать рабочие места с ПК вблизи силовых кабелей и вводов, высоковольтных трансформаторов, технологического оборудования, создающего помехи в работе ПК.

7.2.2. Требования к производственным процессам и оборудованию

Производственные процессы при работе на ПК должны соответствовать требованиям ГОСТ 12.3.002-2014 «ССБТ. Процессы производственные. Общие требования безопасности».

При выполнении работ на ПК согласно ГОСТ 12.0.003-2015 «ССБТ. Опасные и вредные производственные факторы. Классификация» могут иметь место следующие условия:

					ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		62

- повышенная температура корпуса ПК;
- повышенная или пониженная температура воздуха рабочей зоны;
- выделение в воздух рабочего пространства ряда химических веществ;
- повышенная или пониженная влажность воздуха;
- повышенный или пониженный уровень отрицательных и положительных аэроионов;
- повышенное значение напряжения в электрической цепи, замыкание;
- повышенный уровень статического электричества;
- повышенный уровень электромагнитных излучений;
- повышенная напряженность электрического поля;
- отсутствие или недостаток естественного света;
- недостаточная искусственная освещенность рабочей зоны;
- повышенная яркость света;
- повышенная контрастность;
- прямая и отраженная блескость;
- зрительное напряжение;
- монотонность трудового процесса;
- нервно-эмоциональные перегрузки.

Рабочие места с ПК размещаются в изолированных помещениях.

При выполнении работы со значительным умственным напряжением рабочие места рекомендуется изолировать друг от друга специальной перегородкой высотой 1,5...2 м.

Все виды оборудования должны иметь гигиенический сертификат, который включает оценку визуальных параметров.

Конструкция оборудования, его дизайн, эргономические параметры должны обеспечивать надежное и комфортное считывание отображаемой информации.

					ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		63

Чтобы обеспечить надежное считывание информации при соответствующей степени комфортности ее восприятия должны быть определены оптимальные и допустимые диапазоны визуальных эргономических параметров.

При проектировании и разработке ПК сочетания визуальных эргономических параметров и их значения, которые соответствуют оптимальным и допустимым диапазонам, полученных в результате испытаний в специальных лабораториях, аккредитованных в установленном порядке, и подтвержденных соответствующими протоколами, должны быть внесены в техническую документацию на ПК.

При работе с ПК следует обеспечивать значения визуальных параметров в пределах оптимального диапазона; разрешается кратковременная работа при допустимых значениях визуальных параметров. Оптимальные и допустимые значения визуальных эргономических параметров должны быть указаны в технической документации ПК для режимов работы пользователей. При отсутствии в технической документации на ПК данных об оптимальных и допустимых диапазонах значений эргономических параметров эксплуатация ПК не допускается.

Конструкция ПК должна предусматривать наличие регулировок яркости и контрастности от минимальных до максимальных значений.

Конструкция клавиатуры должна предусматривать:

- исполнение в виде обособленного устройства с возможностью свободного размещения;
- опорное приспособление, которое позволяет изменять угол наклона поверхности клавиатуры в пределах от 0 до 10 градусов;
- выделение цветом, размером, формой и местом расположения функциональных групп клавиш.

7.2.3. Требования к отоплению, вентиляции и кондиционированию воздуха

Для предотвращения неблагоприятного воздействия на пользователей в производственных зданиях и рабочих помещениях неблагоприятных факторов, а именно: повышенной (или пониженной) температуры, повышенной относительной влажности и скорости движения воздуха, запыленности, загазованности – следует предусматривать системы отопления, теплоснабжения, вентиляции и кондиционирования воздуха.

Устройство, содержание и эксплуатация систем должны соответствовать требованиям:

- СНиП 41-01-2003 «Отопление, вентиляция и кондиционирование»;
- ГОСТ 12.4.021-75 «ССБТ. Системы вентиляционные. Общие требования (с Изменением 1 от 08.2001)».

Отопительные приборы размещают в местах, которые доступны для осмотра и ремонта, а также очистки, на расстоянии 100 мм от поверхностей стен.

Не разрешается размещать отопительные приборы в нишах стен.

Вентиляция и кондиционирование воздуха должны обеспечивать допустимые и оптимальные нормы микроклиматических параметров на рабочих местах, оснащенных ПК, и содержание вредных веществ в воздухе рабочей зоны не выше ПДК.

Концентрация вредных веществ в приточном воздухе не должна превышать 0,3 ПДК.

Вентиляция производственных зданий в районах с температурой наружного воздуха минус 40 град. С и не ниже должна быть с искусственным отоплением.

Расчет воздухообмена следует проводить по тепло избыткам от машин, людей, солнечной радиации и искусственного освещения.

									Лист
									65
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ				

7.2.4. Требования к организации рабочих мест

Рабочие места должны отвечать требованиям СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы», разделы «Общие требования к организации рабочих мест пользователей ПЭВМ» и «Требования к организации и оборудованию рабочих мест с ПЭВМ для взрослых пользователей».

Работа с применением персональных ПК сопряжена со значительными зрительными и нервно-психологическими нагрузками, что повышает требования к организации труда пользователей ПК (ПЭВМ).

При размещении рабочих мест с ПК расстояние между рабочими столами с мониторами (в направлении тыла поверхности одного монитора и экрана другого монитора), должно быть не менее 2 м, а расстояние между боковыми поверхностями мониторов – не менее 1,2 м.

Рабочие места с ПК в помещениях с источниками вредных производственных факторов размещаются в изолированных кабинах с организованным воздухообменом.

Рабочие места с ПК при выполнении творческой работы рекомендуется изолировать друг от друга перегородками высотой 1,5...2,0 м.

Экран монитора должен располагаться от глаз пользователя на расстоянии 600...700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей, характера выполняемой работы. При этом допускается использование столов различных конструкций, которые отвечают современным требованиям эргономики. Поверхность рабочего стола должна иметь коэффициент отражения 0,5...0,7.

									Лист
									66
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ				

Конструкция рабочего стула (кресла) обязана обеспечивать поддержание рациональной рабочей позы при работе на ПК, позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предотвращения развития утомления. Тип рабочего стула (кресла) следует выбирать с учетом роста пользователя, характера и продолжительности работы с ПК.

Рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Поверхность сиденья, спинки и других элементов стула (кресла) должна быть полумягкой, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, который бы обеспечивал легкую очистку от загрязнений.

Высота рабочей поверхности стола должна регулироваться в пределах 680...800 мм; при отсутствии такой возможности высота рабочей поверхности стола должна быть 725 мм.

Модульными размерами рабочей поверхности стола для ПК, на основании которых рассчитываются конструктивные размеры, следует считать: ширину 800, 1000, 1200 и 1400 мм, глубину 800 и 1000 мм при нерегулируемой его высоте, равной 725 мм.

Рабочий стол должен иметь свободное место для ног высотой не менее 600 мм, шириной – не менее 500 мм, глубиной на уровне колен – не менее 450 мм и на уровне вытянутых ног – не менее 650 мм.

Конструкция рабочего стула должна обеспечивать:

- ширину и глубину поверхности сиденья не менее 400 мм;
- поверхность сиденья с закругленным передним краем;

- регулировку высоты поверхности сиденья в пределах 400...550 мм и углам наклона вперед до 15° и назад до 5°;
- высоту опорной поверхности спинки 300 ± 20 мм, ширину – не менее 380 мм и радиус кривизны горизонтальной плоскости – 400 мм;
- угол наклона спинки в вертикальной плоскости в пределах $\pm 30^\circ$;
- регулировку расстояния спинки от переднего края сиденья в пределах 260...400 мм;
- стационарные или съемные подлокотники длиной не менее 250 мм и шириной – 50...70 мм;
- регулировку подлокотников по высоте над сиденьем в пределах 230 ± 30 мм и внутреннего расстояния между подлокотниками в пределах 350...500 мм.

Рабочее место пользователя ПК рекомендуется оборудовать подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20°. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

Клавиатуру следует располагать на поверхности стола на расстоянии 100...300 мм от края, обращенного к пользователю, или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

7.2.5 Требования к естественному и искусственному освещению

Помещения с ПК должны иметь естественное и искусственное освещение, соответствующее требованиям СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы», разделу «Требования к освещению на рабочих местах, оборудованных ПЭВМ».

					ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		68

Рабочие столы следует размещать таким образом, чтобы мониторы были ориентированы боковой стороной к окнам, чтобы естественный свет падал преимущественно слева.

Искусственное освещение в помещениях с ПК должно осуществляться системой общего равномерного освещения. В производственных и административно-общественных помещениях, в большей степени работы с документами, следует применять системы комбинированного освещения.

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300...500 лк. Освещение не должно создавать бликов на поверхности монитора. Освещенность поверхности монитора не должна быть более 300 лк.

Следует ограничивать прямую блескость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения, должна быть не более 200 кд/м².

Следует ограничивать отраженную блескость на рабочих поверхностях за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране ПК не должна превышать 40 кд/м² и яркость потолка не должна быть больше 200 кд/м².

Показатель ослеплённости для источников общего искусственного освещения в производственных помещениях должно не превышать 20.

Яркость светильников общего освещения в зоне углов излучения от 50 до 90° с вертикалью в продольной и поперечной плоскостях не должна превышать 200 кд/м², защитный угол светильников должен быть более 40°.

Светильники местного освещения должны иметь непросвечивающий отражатель с защитным углом не менее 40°.

Следует ограничивать неравномерность распределения яркости в поле зрения пользователя ПК, при этом соотношение яркости между рабочими

					ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		69

	энерготрат, Вт				м/с, более	не
Холодный	Ia (до 139)	22...24	21...25	60...40	0,1	
	Iб (140...174)	21...23	20...24	60...40	0,1	
Теплый	Ia (до 139)	23...25	22...26	60...40	0,1	
	Iб (140...174)	22...24	21...25	60...40	0,1	

Перепады температуры воздуха по высоте от уровня пола (0,1; 1,0; 1,5) м, а также изменения температуры воздуха в течение рабочего дня при обеспечении оптимальных величин микроклимата на рабочих местах не должны превышать 2°С и выходить за пределы величин, указанных в таблице 7.1.

Воздух в помещении должен соответствовать нормам по содержанию вредных химических веществ, а также аэронов. Уровень ионизации воздуха согласно СанПиН 2.2.4.1294-03 «Гигиенические требования к аэроионному составу воздуха производственных и общественных помещений» представлен в таблице 7.2.

Таблица 7.2 – Уровни ионизации воздуха в помещениях

Нормируемые показатели	Число ионов в 1 см ³ воздуха	
	n+	n-
Минимально допустимое	400	600
Максимально допустимые	50000	50000

Установленный в производственном помещении кондиционер установлен на поддержание оптимальной температуры и влажности.

В зонах дыхания персонала на рабочих местах, где имеются источники электростатических полей (мониторы или другие виды оргтехники) допускается отсутствие аэроионов положительной полярности.

Для обеспечения комфортных условий используются как организационные методы (рациональная организация проведения работ в зависимости от времени года и суток, чередование труда и отдыха), так и технические средства (вентиляция, кондиционирование воздуха, отопительная система).

7.2.7 Требования к шуму и вибрации

Уровень шума и вибрации на рабочих местах, оборудованных ПК должен соответствовать требованиям СанПиН 2.2.4.3359-16 «Санитарно-эпидемиологические требования к физическим факторам на рабочих местах», разделы «Шум на рабочих местах» и «Вибрация на рабочих местах».

Производственные помещения, в которых для работы используются ПК, не должны находиться рядом с помещениями, в которых уровни шума и вибрации превышают нормируемые значения (печатные, механические цеха).

Нормативным эквивалентным уровнем звука на рабочих местах, является 50 дБА.

При сокращенном рабочем дне (менее 40 ч в неделю) предельно допустимые уровни применяются без изменения.

Предельно допустимые величины эквивалентного скорректированного виброускорения за рабочий день производственной вибрации приведены в таблице 7.3.

При сокращенном рабочем дне (менее 40 ч в неделю) ПДУ применяется без изменения.

Таблица 7.3 – Предельно допустимые значения и уровни производственной вибрации

Вид вибрации	Категория вибрации	Направление действия	Коррекция	Нормативные эквивалентные значения и уровни виброускорения	
				м/с ²	дБ
Общая	Зв	Z _o	W _k	0,014	83
		X _o , Y _o	W _d	0,0099	80

7.2.8 Требования к ионизирующим и неионизирующим излучениям

Значительная часть ученых считают, что как кратковременное, так и длительное воздействие всех видов излучения от экрана монитора не опасно для здоровья пользователей компьютеров. Однако исчерпывающих данных

относительно опасности воздействия излучения от мониторов на пользователей компьютеров не существует и исследования в этом направлении продолжаются.

Допустимые значения параметров неионизирующих электромагнитных излучений от монитора компьютера представлены в таблице 7.4.

Таблица 7.4 – Допустимые значения параметров неионизирующих электромагнитных излучений (в соответствии с СанПиНом 2.2.2/2.4.1340-03)

Наименование параметра		Допустимые значения
Напряженность электрического поля	В диапазоне частот от 5 Гц до 2 кГц	25,0 В/м
	В диапазоне от 2 кГц до 400 кГц	2,5 В/м
Плотность магнитного потока	В диапазоне частот от 5 Гц до 2 кГц	250,0 нТл
	В диапазоне от 2 кГц до 400 кГц	25,0 нТл
Напряженность электростатического поля		15,0 кВ/м

Для снижения воздействия этих видов излучения рекомендуется применять мониторы с пониженным уровнем излучения, соблюдать регламентированные режимы труда и отдыха.

7.2.9 Пожарная безопасность

Пожар – это горение вне специального очага, наносящий материальный ущерб и создающий опасность для жизни людей. Нормативным документом пожарной безопасности является ГОСТ 12.1.004-91 ССБТ «Пожарная безопасность. Общие требования (с Изменением 1 от 09.2006)». Пожарная безопасность – состояние защищенности личности и его имущества от процессов неконтролируемого горения.

Снижение пожарной опасности электротехнических изделий и их частей достигается:

- исключением использования в конструкции компонентов легковоспламеняющихся материалов;
- ограничением массы горючих материалов, а также их заменой на более нагревостойкие;

- ограничением проникновения горючих материалов извне к пожароопасным узлам электротехнических изделий;
- применением конструкции изделий, обеспечивающих предотвращение выброса раскаленных и горящих частиц;
- ограничением температуры возможных источников зажигания и выбором режима работы электротехнических компонентов;
- применением средств и элементов, предназначенных для автоматического отключения изделия в аварийном режиме работы и исключающих возгорание частей изделий, выполненных из электроизоляционных материалов.

Учебно-лабораторный комплекс имеет токопроводящие части, которые в процессе эксплуатации выделяют тепло и могут быть причиной воспламенения. Вследствие этого, для рассеивания тепла установлены алюминиевые радиаторы. Экспериментально установлено, что их температура при длительной эксплуатации комплекса не превышает 55 градусов. При касании к ним не возникает ожоговых ощущений. В состав комплекса входят трудногорючие вещества и материалы, которые препятствуют самовозгоранию и поддержанию пламени. Применяется изоляция, что уменьшает возможность соединения токоведущих частей с последующим разогревом до критических температур.

В комплексе используется оборудование, при эксплуатации которого не образуется источников зажигания и не возникает искр.

Также устанавливаются требования к помещению, где эксплуатируется комплекс. В процессе эксплуатации не требуется располагать его в определенных изолированных помещениях или на открытых площадках. Согласно технической документации на комплекс в комнате должны быть установлены огнетушители, а также графически изображены план эвакуации и инструкция, в которой описаны действия при пожаре.

					ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	<i>Лист</i>
						74
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

7.3 Требования к организации режима труда и отдыха

Режимы труда и отдыха при работе на ПК обязаны организовываться в зависимости от вида и категории трудовой деятельности.

Выделяют три группы деятельности:

- группа А – деятельность по считыванию информации с предварительным запросом;
- группа Б – деятельность по вводу информации;
- группа В – творческая деятельность в режиме диалога с ПК.

При выполнении разных групп работ в течение смены за основную принимают такую, которая занимает не менее 50% времени рабочего дня.

Для групп трудовой деятельности различают три категории тяжести и напряженности работы.

Для группы А до 20000 знаков – I категория; до 40000 знаков – II категория; до 60000 знаков – III категория. Не допускается набор более 60000 знаков;

Для группы Б – до 15000 знаков - I категория; до 30000 знаков – II категория; до 40000 знаков – III категория. Не допускается набор более 40000 знаков;

Для группы В – до двух часов – I категория; до четырех часов – II категория; до шести часов – III категория. Не допускается набор более шести часов.

Действующим законодательством о труде и условиями внутреннего трудового распорядка предприятия определяется продолжительность обеденного перерыва.

Для обеспечения наилучшей работоспособности и сохранения здоровья пользователей на протяжении рабочего дня должны устанавливаться регламентированные перерывы.

					ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		75

Время регламентированных перерывов в течение рабочего дня устанавливается в зависимости от ее продолжительности, вида и категории работы.

Суммарное время регламентированных перерывов принимается для групп.

Для групп А, Б и В:

- категория I – 30 мин. (при восьмичасовом рабочем дне) и 70 мин. (при двенадцатичасовом рабочем дне);
- категория II – 50 и 90 мин. соответственно;
- категория III – 70 и 120 мин. соответственно.

Продолжительность непрерывной работы с ПК не должна превышать двух часов.

При выполнении работы на ПК в ночную смену (с 22 часов), независимо от категории и вида трудовой деятельности, длительность регламентированных перерывов должна увеличиваться на один час.

При восьмичасовой смене регламентированные перерывы следует устанавливать следующим образом:

- для I категории работ – через два часа от начала работы и через полтора-два часа после обеденного перерыва длительностью 15 минут каждый;
- для II категории работ – через два часа от начала смены и через полтора-два часа после обеда длительностью 15 мин. каждый или по 10 минут через каждый час работы;
- для III категории работ – через полтора-два часа от начала смены и через полтора-два часа после обеда длительностью 20 минут каждый или по 15 минут через каждый час работы.

При двенадцатичасовой смене регламентированные перерывы должны устанавливаться в первые восемь часов работы аналогично перерывам при

									Лист
									76
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ				

восьмичасовой смене, а в течение последних четырех часов работы, независимо от категории и вида работ, - каждый час продолжительностью 15 минут.

Во время регламентированных перерывов с целью сохранения высокой работоспособности необходимо выполнять комплекс упражнений.

С целью уменьшения отрицательного влияния монотонности необходимо чередовать операции осмысленного текста и числовых данных, чередовать редактирования текстов и ввода данных.

В случаях появления у пользователей ПК зрительного дискомфорта и других негативных субъективных ощущений при соблюдении санитарно-гигиенических, эргономических условий следует применять индивидуальный подход в ограничении времени работ с ПК, коррекцию продолжительности перерывов для отдыха или проводить смену деятельности на другую, не связанную с использованием ПК.

Работающим на ПК с высоким уровнем умственного напряжения во время перерывов и в конце рабочего дня показан отдых в комнатах специальной психологической разгрузки.

13 элементов стула полумягкая, с нескользящим, недостаточно электризующимся и воздухопроницаемым покрытием, которое обеспечивает легкую очистку от загрязнений.

Высота рабочей поверхности стола составляет составлять 725 мм.

Рабочий стол имеет пространство для ног высотой 600 мм и более, шириной – 500 мм и более, глубиной на уровне колен – 450 мм и более, на уровне вытянутых ног - 650 мм и более.

Помещение имеет как естественное, так и комбинированное общее равномерное искусственное освещение. Присутствует одностороннее боковое естественное освещение. Искусственное освещение осуществляется системой общего равномерного освещения люминесцентных двухламповых источников типа ЛБ в количестве 8 штук, каждая из которых мощностью 58 Вт. На каждом рабочем месте

									Лист
									77
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ				

присутствуют светильники местного освещения, в качестве источников применяются люминесцентные лампы мощностью 11 Вт.

В помещении имеется отопление, вентиляция.

Основными источниками шума являются системные блоки ПК и МФУ.

Рабочие места с ПК оборудованы защитным занулением в соответствии техническими требованиями по эксплуатации.

На рисунке 7.1 показана схема размещения рабочих мест.

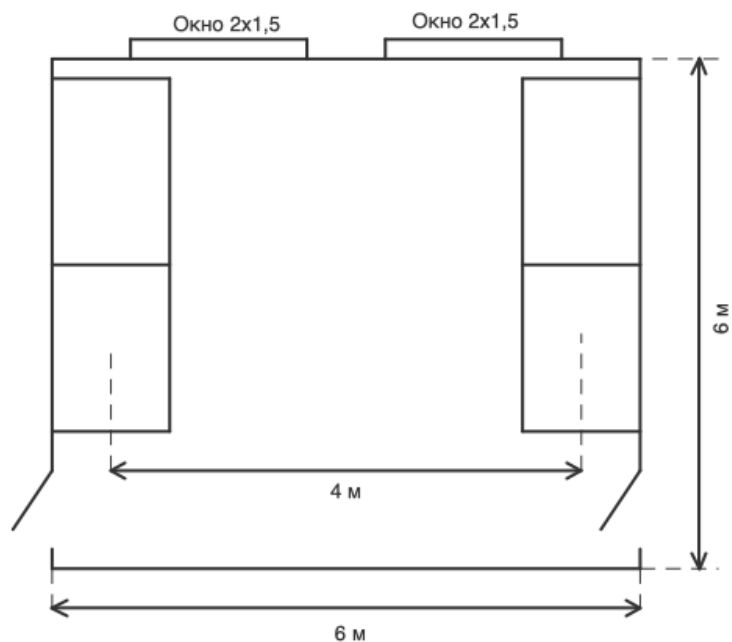


Рисунок 7.1 – Схема размещения рабочих мест в помещении

ЗАКЛЮЧЕНИЕ

В результате выполнения данного дипломного проекта было разработано программное обеспечение, предназначенное для управления стендом имитации ультразвуковой РЛС. Компьютерное программное обеспечение работает на операционной системе Astra Linux и решает такие задачи, как: определение расстояния цели импульсным, фазовым и частотным методами, определение угловых координат цели амплитудным и фазовым методами, измерение скорости цели доплеровским методом.

Планируется дальнейшее развитие разработки, усовершенствование и оптимизация алгоритмов программной части управляющего контроллера, реализация дополнительного функционала, например, имитация процесса секторного обзора с подвижными целями, автоматическое слежение за подвижной целью по угловым координатам.

В организационно-экономическом разделе рассчитана экономическая эффективность разработанного программного обеспечения.

Разработаны правила охраны труда для обеспечения безопасности пользователей персонального компьютера, входящего в состав учебного комплекса.

									Лист
									79
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ				

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Донской А. В., Келлер О. К., Кратыш Г. С. – Ультразвуковые электротехнологические установки. – Л.: Энергоиздат., 1982. – 208 с.
2. Бабилов О. И. – Ультразвук и его применение в промышленности. – М.: Наука, 1958. - 260 с.: ил.
3. Жемеров Д., Исакова С. Kotlin в действии / пер. с англ. Киселев А.Н. – М.: ДМК Пресс, 2018. – 402 с.
4. Григорин-Рябов В.В., Васин В.В. Радиолокационные устройства (теория и принципы построения). – М.: «Советское радио», 1970. – 680 с.
5. Неганов В.А., Ключев Д.С., Табаков Д.П. Устройства СВЧ и антенны. Теория и техника антенн. – М.: URSS, 2013. – 728 с.
6. Хансен Р.С. Фазированные антенные решетки. – М.: Техносфера, 2012. – 560 с.
7. Под ред. П.И. Дудника – Авиационные радиолокационные комплексы и системы: учебник для слушателей и курсантов ВУЗов ВВС. – М.: Изд. ВВИА им. проф. Н.Е. Жуковского, 2006. – 1112 с.
8. Стивен П. – Язык программирования C++. Лекции и упражнения. – М.: Вильямс, 2012. – 1248 с.: ил.
9. Романов Е. Л. – Практикум по программированию на C++. – М.: БХВ-Петербург, 2004. – 432 с.: ил.
10. Электронный ресурс <http://ru.scribd.com/doc/252234413/Инсайдерское-руководство-по-STM32> (Internet)
11. Электронный ресурс <http://avr.ru/docs/books/arm> (Internet)
12. Электронный ресурс Datasheet STM32F407VGT6: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00039084.pdf (Internet)

									Лист
									80
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ				

13. Мешковой Н.П., Закиров Р.Ш., Зинкевич В.С., Попов С.Г. – Учебное пособие для студентов приборостроительного факультета. – Челябинск: ЧГТУ, 1990. – 53 с.

14. Скворцов Ю.В. – Организационно-экономические вопросы в дипломном проектировании – М.: Студент, 2012. – 374 с.: ил.

15. Электронный ресурс Windows 10 – Microsoft Store: <https://www.microsoft.com/ru-ru/store/b/windows> (Internet)

16. Зотов Б.И. – Безопасность жизнедеятельности на производстве: учебник. – 2-е изд., перераб. и доп. – Изд-во: КолосС, 2004. – 432 с.: ил.

17. Бекасова В.Н., Боровик С.И., Глотова Н.В., Давлятшин В.Г., Зеленкин В.Г., Киселева Л.М., Окраинская И.С. – Безопасность жизнедеятельности в дипломных проектах. – Челябинск: ЮУрГУ, 2007. – 166 с.: ил.

18. СТО ЮУрГУ 04 – 2008: Курсовое и дипломное проектирование: Общие требования к оформлению. – Челябинск: ЮУрГУ, 2008. – 56 с.

19. ГОСТ 2.001-93 ЕСКД. «Общие положения» Москва: ИПК Издательство стандартов.

20. ГОСТ 2.301-68 ЕСКД. «Форматы» Москва: ИПК Издательство стандартов.

21. ГОСТ 2.302-68 ЕСКД. Масштабы. Москва: ИПК Издательство стандартов.

22. ГОСТ 2.303-68 ЕСКД. Линии. Москва: ИПК Издательство стандартов.

23. ГОСТ 2.304-81 ЕСКД. Шрифты чертёжные. Москва: ИПК Издательство стандартов.

24. ГОСТ 2.701-2008 ЕСКД. Схемы. Виды и типы. Общие требования к выполнению. Москва: ИПК Издательство стандартов.

									Лист
									81
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ				

ПРИЛОЖЕНИЕ А

Исходный код программы для микроконтроллера

```
#include "stm32f4xx.h"//подключение библиотеки описания контроллера

uint16_t i,j,k,m,n; //Переменные-счетчики для циклов

//массивы принятых с АЦП значений
uint16_t ADCValue1[7680]; //С АЦП№1
uint16_t ADCValue2[7680]; //С АЦП№2
//*****Объявление функций*****//
void simpleDelay(uint32_t delayTime) //функция задержки
{
    uint32_t delays;
    for (delays = 0; delays < delayTime; delays++);
}

void Max_Speed_Clk(void) //функция разгона STM32F407VGT6 до 168МГц
{
    #define PWR_CR_VOS_Bit (*((__IO uint32_t *) (0x42000000 + (0x00007000 * 0x20) +
(14*4))))
    #define RCC_CR_HSEON_Bit (*((__IO uint32_t *) (0x42000000 + ((0x40023800 &
0x00FFFFFF) * 0x20) + (16*4))))
    #define RCC_CR_PLLON_Bit (*((__IO uint32_t *) (0x42000000 + ((0x40023800 &
0x00FFFFFF) * 0x20) + (24*4))))
    #define RCC_CFGR_SW_PLL_Bit (*((__IO uint32_t *) (0x42000000 + ((0x40023808 &
0x00FFFFFF) * 0x20) + (1*4))))
    #define RCC_APB1ENR_PWREN_Bit (*((__IO uint32_t *) (0x42000000 + ((0x40023840 &
0x00FFFFFF) * 0x20) + (28*4))))
    #define RCC_PLLCFGR_PLLSRC_HSE_Bit (*((__IO uint32_t *) (0x42000000 +
((0x40023804 & 0x00FFFFFF) * 0x20) + (22*4))))
    #define PLL_M 24 //Частота внешнего кварца
    #define PLL_N 336 //Значение делителя ФАПЧ
    #define PLL_P 2 //Значение делителя после ФАПЧ перед
коммутатором системной частоты
    #define PLL_Q 7 //Значение делителя после ФАПЧ для шин USB
    OTG FS, SDIO, RNG

    RCC_APB1ENR_PWREN_Bit = 1; //Включение тактирования интерфейса
    PWR_CR_VOS_Bit = 1; //Режим №1 регулятора напряжения
    RCC_CR_HSEON_Bit = 1; //Включение тактирования от внешнего
кварца
    while (!(RCC->CR & RCC_CR_HSERDY)); //Ожидание уст. тактирования от
кварца
    RCC->CFGR |= RCC_CFGR_HPRE_DIV1; //Значение предделителя АНВ
    RCC->CFGR |= RCC_CFGR_PPRE2_DIV2; //Значение предделителя APB2
    RCC->CFGR |= RCC_CFGR_PPRE1_DIV4; //Значение предделителя APB1
    RCC->PLLCFGR = PLL_M | (PLL_N << 6) | (((PLL_P >> 1) - 1) << 16)|
(RCC_PLLCFGR_PLLSRC_HSI) | (PLL_Q << 24);
```

						Лист
					ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	82
Изм.	Лист	№ докум.	Подпись	Дата		

```

RCC_PLLCFGR_PLLSRC_HSE_Bit = 1;
RCC_CR_PLLON_Bit = 1;
while ((RCC->CR & RCC_CR_PLLRDY) == 0);
FLASH->ACR = FLASH_ACR_ICEN | FLASH_ACR_DCEN |
FLASH_ACR_LATENCY_5WS;
RCC->CFGR &= ~(RCC_CFGR_SW);
RCC_CFGR_SW_PLL_Bit = 1;
while ((RCC->CFGR & (uint32_t)RCC_CFGR_SWS) != RCC_CFGR_SWS_PLL);
}

void Error_Handler(void)
{
    while(1)
    {
    }
}

uint16_t Buffercmp(uint8_t* pBuffer1, uint8_t* pBuffer2, uint16_t BufferLength)
{
    while (BufferLength--)
    {
        if((*pBuffer1) != *pBuffer2)
        {
            return BufferLength;
        }
        pBuffer1++;
        pBuffer2++;
    }
    return 0;
}

void DMA_Config(void)
{
    /*## -1- Enable DMA2 clock */
    HAL_RCC_DMA2_CLK_ENABLE();
    /*##-2- Select the DMA functional Parameters */
    DmaHandle.Init.Channel = DMA_CHANNEL; /* DMA_CHANNEL_0 */
    DmaHandle.Init.Direction = DMA_MEMORY_TO_MEMORY; /* M2M transfer mode */
    DmaHandle.Init.PeriphInc = DMA_PINC_ENABLE; /* Peripheral increment mode Enable */
    DmaHandle.Init.MemInc = DMA_MINC_ENABLE; /* Memory increment mode Enable */
    DmaHandle.Init.PeriphDataAlignment = DMA_PDATAALIGN_WORD; /* Peripheral data
alignment : Word */
    DmaHandle.Init.MemDataAlignment = DMA_MDATAALIGN_WORD; /* memory data
alignment : Word */
    DmaHandle.Init.Mode = DMA_NORMAL; /* Normal DMA mode */
    DmaHandle.Init.Priority = DMA_PRIORITY_HIGH; /* priority level : high */
    DmaHandle.Init.FIFOMode = DMA_FIFOMODE_ENABLE; /* FIFO mode enabled */
    DmaHandle.Init.FIFOThreshold = DMA_FIFO_THRESHOLD_1QUARTERFULL; /* FIFO
threshold: 1/4 full */
    DmaHandle.Init.MemBurst = DMA_MBURST_SINGLE; /* Memory burst */
    DmaHandle.Init.PeriphBurst = DMA_PBURST_SINGLE; /* Peripheral burst */
}

```

```

/*##-3- Select the DMA instance to be used for the transfer : DMA2_Stream0 */
DmaHandle.Instance = DMA_STREAM;

/*##-4- Initialize the DMA stream */
if(HAL_DMA_Init(&DmaHandle) != HAL_OK)
{
    /* Initialization Error */
    Error_Handler();
}

/*##-5- Select Callbacks functions called after Transfer complete and Transfer error */
HAL_DMA_RegisterCallback(&DmaHandle, HAL_DMA_XFER_CPLT_CB_ID,
TransferComplete);
HAL_DMA_RegisterCallback(&DmaHandle, HAL_DMA_XFER_ERROR_CB_ID,
TransferError);

/*##-6- Configure NVIC for DMA transfer complete/error interrupts */
/* Set Interrupt Group Priority */
HAL_NVIC_SetPriority(DMA_STREAM_IRQ, 0, 0);
/* Enable the DMA STREAM global Interrupt */
HAL_NVIC_EnableIRQ(DMA_STREAM_IRQ);
/*##-7- Start the DMA transfer using the interrupt mode #####*/
/* Configure the source, destination and buffer size DMA fields and Start DMA Stream
transfer */
/* Enable All the DMA interrupts */
if(HAL_DMA_Start_IT(&DmaHandle, (uint32_t)&aSRC_Const_Buffer,
(uint32_t)&aDST_Buffer, BUFFER_SIZE) != HAL_OK)
{
    /* Transfer Error */
    Error_Handler();
}
}

```

ПРИЛОЖЕНИЕ Б

Исходный код программы персонального компьютера

```
/**
 * Лончер приложения.
 *
 * Параметры запуска:
 * * `--debug` - режим отладки, в консоль будет выводиться отладочная информация
 * * `--usb-tools` - запускает только подпрограмму по работе с USB устройствами напрямую
 *
 * @param args аргументы
 */
fun main(args: Array<String>) {
    App.start(args)
}

/**
 * Singleton приложения, является ядром запуска всех модулей и подпрограмм приложения.
 * Также является источником зависимостей приложения.
 */
object App : KodeinAware {

    // Модуль зависимостей ядра
    override val kodein: Kodein = Kodein.lazy {
        bind<App>() with instance(this@App)
        import(appModule)
    }
    // Триггер вреднения зависимостей
    override val kodeinTrigger = KodeinTrigger()

    // Флаг отладки
    var debug: Boolean = false
        private set

    // Строковые ресурсы
    private val strings: ResourceBundle by instance(STRINGS)

    /**
     * Запускает приложение с заданными параметрами
     *
     * @param args параметры запуска приложения
     */
    fun start(args: Array<String>) {
        Log.info("App::start(args = '${args.joinToString(" ")}')")

        val optionDebug = Option.builder()
            .longOpt("debug")
            .required(false)
            .desc(strings.string("args.debug.description"))
    }
}
```

									Лист
									85
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ				

```

        .build()
    val usbToolOption = Option.builder()
        .longOpt("usb-tools")
        .required(false)
        .desc(strings.string("args.usbtools.description"))
        .build()

    val options = Options()
        .addOption(optionDebug)
        .addOption(usbToolOption)

    val parser = DefaultParser()

    kodeinTrigger.trigger()

    try {
        val commandLine = parser.parse(options, args)

        // Если указан параметр режима отладки
        debug = commandLine.hasOption(optionDebug.longOpt)
        // Включаем или отключаем журналирование в зависимости от режима отладки
        Log.enableDebug(debug)

        // Если указан параметр для запуска подпрограммы работы с USB
        if (commandLine.hasOption(usbToolOption.longOpt)) {
            // то запускаем подпрограмму
            Application.launch(UsbToolsApp::class.java)
        } else {

        }
    } catch (exception: ParseException) {
        System.err.println("Ошибка обработки параметров. Причина: ${exception.message}")
    }
}

/**
 * Интерфейс базового контракта экрана
 */
interface BaseContract {

    /**
     * Интерфейс отображения
     */
    interface View<P> {

        /**
         * Ссылка на презентер
         */
        val presenter: P
    }
}

```

```

/**
 * Событие создания отображения
 */
fun onCreate()

/**
 * Событие уничтожения отображения
 */
fun onDestroy()
}

/**
 * Интерфейс презентера
 */
interface Presenter<V> {

    /**
     * Ссылка на тображение
     */
    val view: V

    /**
     * Событие создания отображения (в том числе презентера)
     */
    fun onCreate()

    /**
     * Событие уничтожения отображения (в том числе презентера)
     */
    fun onDestroy()
}
}

/**
 * Базовая реализация отображения. В классе реализуется внедрение зависимостей через
 [KodeinAware], а также привязка
 * к жизненному циклу JavaFX приложения через [Initializable].
 */
abstract class BaseView<P : BaseContract.Presenter<*>> : BaseContract.View<P>, KodeinAware,
Initializable {

    // Создаем собственный модуль зависимостей, включающий в себя зависимости ядра
    override val kodein: Kodein = Kodein.lazy {
        extend(App.kodein)
    }
    // Создаем собственный контекст зависимостей, привязанный к текущему объекту
    override val kodeinContext: KodeinContext<out BaseView<P>>
        get() = kcontext(this)

    /**

```

						Лист
						87
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	

```

* Инициализация объекта, после создания представления JavaFX и биндинга всех
элементов через аннотацию [FXML].
*/
final override fun initialize(location: URL?, resources: ResourceBundle?) {
    onCreate()
}

override fun onCreate() {
    // Передаем событие презентеру о создании отображения
    presenter.onCreate()
}

override fun onDestroy() {
    // Передаем событие презентеру об уничтожении отображения
    presenter.onDestroy()
}
}

/**
* Базовая реализация презентера.
* В классе определяется внедрение зависимостей через [KodeinAware], а также реализация
скоуп выполнения корутин
* [CoroutineScope], позволяющий корутинам привязываться к жизненному циклу презентера.
*/
abstract class BasePresenter<V> : BaseContract.Presenter<V>, KodeinAware, CoroutineScope {
    /**
    * Job презентера. Все корутины, выполняемые в контексте презентера являются дочерними
и также привязываются к
    * жизненному циклу презентера. Отменяя работу текущей Job, отменяется работа всех
дочерних Job.
    */
    private lateinit var job: Job
    /**
    * Контекст выполнения дочерних корутин. Включает [job] и основной поток JavaFX
приложения.
    */
    override val coroutineContext: CoroutineContext
        get() = job + Dispatchers.Main

    override fun onCreate() {
        // Создаем Job, отмена работы которого снизу вверх, от дочерних Job'ов, происходить не
будет, только сверху
        // вниз, отменяя текущую Job
        job = SupervisorJob()
    }

    override fun onDestroy() {
        // При уничтожении презентера, отменяем все корутины
        job.cancel()
    }
}

```



```

/**
 * Базовая реализация приложения (подпрограммы).
 * В приложении присутствует 2 этапа жизненного цикла: создание [onCreate] и уничтожение
 [onDestroy].
 */
abstract class BaseApplication : Application(), KodeinAware {

    // Создаем модуль зависимостей и включаем в него зависимости ядра
    override val kodein: Kodein = Kodein.lazy {
        extend(App.kodein)
    }
    // Создаем собственный контекст зависимостей, привязанный к текущему объекту
    override val kodeinContext: KodeinContext<out BaseApplication>
        get() = kcontext(this)

    /**
     * Переопределенный метод, вызывающийся при запуске приложения.
     * Запрещаем его переопределение у наследуемых классов.
     */
    final override fun start(primaryStage: Stage) {
        onCreate(primaryStage)
    }

    /**
     * Переопределенный метод, вызывающийся при завершении работы приложения.
     * Запрещаем его переопределение у наследуемых классов.
     */
    final override fun stop() {
        super.stop()
        onDestroy()
    }

    /**
     * Метод события жизненного цикла, вызывающийся при создании приложения.
     */
    abstract fun onCreate(primaryStage: Stage)

    /**
     * Метод жизненного цикла, вызывающийся при уничтожении (завершении работы)
 приложения.
     */
    abstract fun onDestroy()
}

/**
 * Интерфейс - часть системы обработки ошибки, определяющий функции для отображения
 ошибок пользователю
 */
interface UiResolver {

    /**

```

						Лист
					ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	89
Изм.	Лист	№ докум.	Подпись	Дата		

```

    * Отображает сообщение об ошибке [message]
    */
    fun showMessage(message: String)
}

/**
 * Реализация системы обработки ошибок в слое presentation
 *
 * @property resolver
 */
class UiResolution(
    val resolver: UiResolver
) : Resolution() {

    override fun onGenericException(exception: Throwable) {
        resolver.showMessage(exception.getStackTraceString())
    }
}

/**
 * Класс, реализующий отображение ошибки в виде окошка предупреждения
 */
class AlertResolver(
    private val strings: ResourceBundle
) : UiResolver {

    override fun showMessage(message: String) {
        val alert = Alert(Alert.AlertType.ERROR)
        alert.title = strings.string("error.title")
        alert.headerText = message
        alert.showAndWait()
    }
}

/**
 * Класс, реализующий функции для связи с устройством
 */
class DeviceConnection : Connection {

    private var device: UsbDevice? = null
    private var iface: UsbInterface? = null

    override fun onCreate() {
        // Инициализация библиотеки
        Usb.init()
        // Включаем режим отладки
        Usb.enableDebug(isEnable = App.debug)
    }

    override fun onDestroy() {
        // Деинициализируем
    }
}

```

```

    Usb.destroy()
    // Освобождаем устройство
    releaseDevice()
}

override fun getDeviceList(): List<UsbDevice> = Usb.getDeviceList()

override fun writeAndRead(sentData: ByteArray, sizeOfReceivedData: Int): ByteArray {
    if (device == null) initDevice()
    try {
        iface!!.claim { true }

        val inEndpoint = iface!!.getUsbEndpoint(IN_ENDPOINT) as UsbEndpoint
        val outEndpoint = iface!!.getUsbEndpoint(OUT_ENDPOINT) as UsbEndpoint
        val inPipe = inEndpoint.usbPipe
        val outPipe = outEndpoint.usbPipe

        try {
            outPipe.open()
            outPipe.syncSubmit(sentData)
        } finally {
            outPipe.close()
        }

        val receivedData = ByteArray(sizeOfReceivedData)
        try {
            inPipe.open()
            inPipe.syncSubmit(receivedData)
        } finally {
            inPipe.close()
        }

        return receivedData
    } finally {
        iface!!.release()
    }
}

private fun initDevice() {
    // Открываем устройство
    device = Usb.findDevice(vendorId = VENDOR_ID, productId = PRODUCT_ID)
    iface = Usb.getDeviceInterface(device!!, INTERFACE)
}

private fun releaseDevice() {
    device = null
    iface = null
}
}

```

						Лист
						91
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	

```

/**
 * Объект обертки над библиотекой LibUsb
 */
object Usb {

    fun getDeviceList(): List<UsbDevice> {
        fun findDeviceList(hub: UsbHub, deviceList: MutableList<UsbDevice>) {
            deviceList += hub
            hub.attachedUsbDevices.map { it as UsbDevice }.forEach { device ->
                deviceList += device
                if (device is UsbHub) {
                    findDeviceList(device, deviceList)
                }
            }
        }

        val deviceList = mutableListOf<UsbDevice>()
        val rootHub = getUsbRootHoob() as? UsbHub ?: return deviceList
        findDeviceList(rootHub, deviceList)
        return deviceList
    }

    /**
     *
     */
    fun openDevice(context: Context? = null, vid: Short, pid: Short, interfaceIndex: Int):
    DeviceHandle =
        LibUsb.openDeviceWithVidPid(context, vid, pid)
            ?: throw LibUsbException("Test device not found.", -1)

    /**
     *
     */
    fun claimInterface(handle: DeviceHandle, interfaceIndex: Int) {
        // Запрашиваем доступ к интерфейсу
        val result = LibUsb.claimInterface(handle, interfaceIndex)
        if (result != LibUsb.SUCCESS) {
            throw LibUsbException("Unable to claim interface", result)
        }
    }

    /**
     *
     */
    fun releaseInterface(handle: DeviceHandle, interfaceIndex: Int) {
        // Освобождаем интерфейс
        val result = LibUsb.releaseInterface(handle, interfaceIndex)
        if (result != LibUsb.SUCCESS) {
            throw LibUsbException("Unable to release interface", result)
        }
    }
}

```

```

/**
 *
 */
fun closeDevice(context: Context? = null, handle: DeviceHandle, interfaceIndex: Int) {
    // Закрываем устройство
    LibUsb.close(handle)
}

/**
 * СЧИТЫВАЕТ данные с устройства
 */
fun readData(handle: DeviceHandle, size: Int, inEndpoint: Byte, timeout: Long): ByteArray {
    val buffer = BufferUtils.allocateByteBuffer(size).order(ByteOrder.LITTLE_ENDIAN)
    val transferred = BufferUtils.allocateIntBuffer()
    val result = LibUsb.bulkTransfer(handle, inEndpoint, buffer, transferred, timeout)
    if (result != LibUsb.SUCCESS) {
        throw LibUsbException("Unable to read data", result)
    }
    val array = ByteArray(size)
    buffer.get(array)
    return array
}

/**
 * Отправляет данные на устройство
 */
fun writeData(handle: DeviceHandle, data: ByteArray, outEndpoint: Byte, timeout: Long) {
    val outBuffer = BufferUtils.allocateByteBuffer(data.size)
    outBuffer.put(data)
    val transferred = BufferUtils.allocateIntBuffer()
    val writeResult = LibUsb.bulkTransfer(handle, outEndpoint, outBuffer, transferred, timeout)
    if (writeResult != LibUsb.SUCCESS) {
        throw LibUsbException("Unable to send data", writeResult)
    }
}

/**
 * Отправляет, а потом считывает данные с устройства
 */
fun writeAndReadData(handle: DeviceHandle, sendData: ByteArray, size: Int, inEndpoint: Byte,
outEndpoint: Byte, timeout: Long): ByteArray {
    val outBuffer = BufferUtils.allocateByteBuffer(sendData.size)
    outBuffer.put(sendData)
    val transferred = BufferUtils.allocateIntBuffer()
    val writeResult = LibUsb.bulkTransfer(handle, outEndpoint, outBuffer, transferred, timeout)
    if (writeResult != LibUsb.SUCCESS) {
        throw LibUsbException("Unable to send data", writeResult)
    }

    val inBuffer = BufferUtils.allocateByteBuffer(size).order(ByteOrder.LITTLE_ENDIAN)

```

						Лист
					ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	93
Изм.	Лист	№ докум.	Подпись	Дата		

```

val readResult = LibUsb.bulkTransfer(handle, inEndpoint, inBuffer, transferred, timeout)
if (readResult != LibUsb.SUCCESS) {
    throw LibUsbException("Unable to read data", readResult)
}
val array = ByteArray(size)
inBuffer.get(array)
return array
}

fun getUsbRootHoob(): UsbDevice? {
    try {
        val services = UsbHostManager.getUsbServices()
        return services.rootUsbHub
    } catch (e: SecurityException) {
        e.printStackTrace()
    } catch (e: UsbException) {
        e.printStackTrace()
    }
    return null
}

fun findDevice(vendorId: Short, productId: Short): UsbDevice? =
    (getUsbRootHoob() as? UsbHub)?.let { findDevice(it, vendorId, productId) }

fun findDevice(hub: UsbHub, vendorId: Short, productId: Short): UsbDevice? {
    hub.attachedUsbDevices.map { it as UsbDevice }.forEach { device ->
        val desc = device.usbDeviceDescriptor
        if (desc.idVendor() == vendorId && desc.idProduct() == productId) {
            return device
        }
        if (device.isUsbHub) {
            val childDevice = findDevice(device as UsbHub, vendorId, productId)
            if (childDevice != null) {
                return device
            }
        }
    }
    return null
}

fun getDeviceInterface(device: UsbDevice, index: Int): UsbInterface {
    val configuration = device.activeUsbConfiguration
    return configuration.getUsbInterface(index.toByte()) as UsbInterface
}

/**
 * Перечисление типов направления работы endpoint
 */
enum class Direction {
    /**
     * Направление для чтения
    */
}

```

```

    */
    IN,
    /**
     * Направления для записи
     */
    OUT
}
}

/**
 *
 */
val Device.descriptor: DeviceDescriptor
    get() {
        val descriptor = DeviceDescriptor()
        val result = LibUsb.getDeviceDescriptor(this, descriptor)
        if (result < 0) {
            throw LibUsbException("Unable to read device descriptor", result)
        }
        return descriptor
    }

/**
 *
 */
val Device.address: Int
    get() = LibUsb.getDeviceAddress(this)

/**
 *
 */
val Device.busNumber: Int
    get() = LibUsb.getBusNumber(this)

/**
 *
 */
fun Device.useDevice(block: (handle: DeviceHandle) -> Unit) {
    val handle = DeviceHandle()
    val result = LibUsb.open(this, handle)
    if (result < 0) {
        throw LibUsbException("Unable to open device", result)
    }
    try {
        block(handle)
    } finally {
        LibUsb.close(handle)
    }
}

/**

```

						Лист
						95
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	

```

*
*/
fun Device.useConfigurations(block: (configIndex: Int, configDescriptor: ConfigDescriptor) -> Unit)
{
    val numConfigurations = descriptor.bNumConfigurations().toInt() and 0xff
    var i = 0
    while (i < numConfigurations) {
        val configDescriptor = ConfigDescriptor()
        val result = LibUsb.getConfigDescriptor(this, i.toByte(), configDescriptor)
        if (result < 0) {
            throw LibUsbException("Unable to get configuration $i for device $this", result)
        }

        try {
            block(i, configDescriptor)
        } finally {
            LibUsb.freeConfigDescriptor(configDescriptor)
        }
        i += 1
    }
}

/**
 *
 */
val DeviceDescriptor.vendorId: String
    get() = "0x${UsbUtil.toHexString(idVendor()).toUpperCase()}"

/**
 *
 */
val UsbDeviceDescriptor.vendorId: String
    get() = "0x${UsbUtil.toHexString(idVendor()).toUpperCase()}"

/**
 *
 */
val DeviceDescriptor.productId: String
    get() = "0x${UsbUtil.toHexString(idProduct()).toUpperCase()}"

/**
 *
 */
val UsbDeviceDescriptor.productId: String
    get() = "0x${UsbUtil.toHexString(idProduct()).toUpperCase()}"

/**
 *
 */
fun DeviceDescriptor.getProduct(handle: DeviceHandle): String? =
    LibUsb.getStringDescriptor(handle, iProduct())

```



```

/**
 *
 */
fun DeviceDescriptor.getManufacturer(handle: DeviceHandle): String? =
    LibUsb.getStringDescriptor(handle, iManufacturer())

/**
 *
 */
fun ConfigDescriptor.useInterfaces(block: (interfaceIndex: Int, interfaceDescriptor:
InterfaceDescriptor) -> Unit) {
    val activeSettings = mutableMapOf<Int, InterfaceDescriptor>()
    for (iface in iface()) {
        for (ifaceDescriptor in iface.altsetting()) {
            val ifaceNumber = ifaceDescriptor.bInterfaceNumber().toInt() and 0xff
            if (!activeSettings.containsKey(ifaceNumber) || ifaceDescriptor.bAlternateSetting().toInt() ==
0) {
                activeSettings[ifaceNumber] = ifaceDescriptor
            }
        }
    }
    activeSettings.values.forEachIndexed(block)
}

fun InterfaceDescriptor.useEndpoints(block: (endpointDescriptor: EndpointDescriptor) -> Unit) {
    for (endpoint in endpoint()) {
        block(endpoint)
    }
}

val EndpointDescriptor.direction: Usb.Direction
    get() {
        val address = bEndpointAddress()
        return if (address.toInt() and LibUsb.ENDPOINT_IN.toInt() == 0) Usb.Direction.OUT else
Usb.Direction.IN
    }

val UsbEndpoint.usbDirection: Usb.Direction
    get() = if (direction == UsbConst.ENDPOINT_DIRECTION_IN) Usb.Direction.IN else
Usb.Direction.OUT

val EndpointDescriptor.address: String
    get() {
        val address = bEndpointAddress()
        return "0x${UsbUtil.toHexString(address).toUpperCase()}"
    }

val UsbEndpoint.address: String
    get() = "0x${UsbUtil.toHexString(usbEndpointDescriptor.bEndpointAddress()).toUpperCase()}"

```

						Лист
						97
Изм.	Лист	№ докум.	Подпись	Дата	ЮУрГУ–Д.11.05.01.2019.307.00 ПЗ	