

Министерство науки и высшего образования Российской Федерации
Лысьвенский филиал федерального государственного автономного образовательного
учреждения высшего образования
Пермский национальный исследовательский политехнический университет
Факультет профессионального образования

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему «Разработка проекта «голографических» часов для ИП Пермяков
А.В.»

студента группы КСК9-18-1спо по специальности 09.02.01 Компьютерные
системы и комплексы

Лаптев Максим Андреевич _____

Руководитель: _____ (С.А.Зыкин)

Консультант по экономической части:
_____ (К.В.Кондратьева)

Консультант по промышленной экологии и охране труда:
_____ (А.К.Тороцин)

Рецензент: _____ (_____)

Допуск к защите: _____ (М.Н.Апталаев)

Лысьва, 2022 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ.....	5
1.1 Понятие голограммы и их виды	5
1.2 Принципы работы голографии	9
1.3 Анализ рынка голографических часов.....	11
1.4 Анализ требований к разрабатываемому устройству.....	16
2 КОНСТРУКТОРСКИЙ РАЗДЕЛ	17
2.1 Структурная схема разрабатываемого устройства	17
2.2 Выбор комплектующих для разрабатываемого устройства.....	18
2.3 Разработка программы и устройства	21
3 ОХРАНА ТРУДА И ПРОМЫШЛЕННАЯ ЭКОЛОГИЯ	31
3.1 Анализ вредных и опасных факторов на рабочем месте веб-дизайнера.....	31
3.2 Освещение рабочего места.....	33
3.3 Утилизация средств вычислительной техники и оргтехники	36
3.4 Вывод по разделу	37
4 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКАЯ ЧАСТЬ	39
4.1 Расчет затрат на разработку программы для микроконтроллера.....	39
4.2 Расчет затрат на внедрение программы для микроконтроллера.....	42
4.3 Расчет эксплуатационных текущих затрат по программе для микроконтроллера	44
4.4 Расчет экономической целесообразности разработки и внедрения информационных технологий.....	47
ЗАКЛЮЧЕНИЕ.....	52
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	53
Приложение А – Программный код.....	55

ВВЕДЕНИЕ

Часы – это указатель времени и один из важных интерьерных деталей во все времена. С их помощью люди могут узнать который в данный момент час или минута. Со временем часы стали модифицировать под свои вкусы. Изобретено множество форм: круглые, квадратные, толстые, тонкие. Есть часы размером с горошину, а есть - огромные. Но голографические – особенный вид.

Голография – это особый метод вывода изображения трехмерных объектов, в котором для восприятия объемного изображения не требуются какие-либо приспособления в виде очков и других устройств.

Мы живём в современном мире, где на счету каждая минута. Часы у нас на руке и в кармане, дома и в офисе, в школе и в магазинах, на вокзале, в автомобиле, на самолёте, в телефоне и на компьютерах.

Разработка имеет актуальность для людей, которым нравится тема будущего и ценят свое время. Эти часы являются уникальным элементом декора, от которого ни много ни мало зависит внешний вид помещения и впечатление, которое оно производит на людей. Такие часы можно поставить в офисе или дома. Хотя в наше время большинство использует наручные часы, но все же люди все еще используют настольные или настенные потому, что они дополняют интерьер дома или офиса.

Выпускная квалификационная работа ориентирована на комфорт в повседневном использовании с использованием обмана зрения (иллюзии) и современных технологий.

Главное отличие выпускной квалификационной работы от текущих предложений на рынке заключается, в том, что данная разработка будет более узкоспециализирована для тех, кто хочет получить максимум удовольствия от просмотра времени.

Объект исследования – голографические часы

Предмет исследования – голографическое автоматическое устройство счета текущего времени.

Целью выпускной квалификационной работы является разработка «Голографических часов».

На основании поставленной цели были выявлены следующие задачи:

- провести анализ предметной области исследования и разработать технические требования к проектируемой системе;
- разработать аппаратную и программную часть устройства;
- выполнить технико-экономическое обоснование проекта;
- рассмотреть требования по охране труда и промышленной экологии.

1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

1.1 Понятие голограммы и их виды

Голография – это определённый способ регистрации информации, в результате которого получается объёмное 3-х мерное изображение.

Термин голография был выдвинут ещё в 1948 году, но по причине неразвитости технологий, практически выполнять работы с голографией получилось лишь в 70-е годы.

Голография представляет собой 3-х мерное изображение, расположенное на некоторой стеклянной пластине. Это добивается путём регистрации большей информации, в отличии от той информации, которая фиксируется при выполнении обычной фотографии [1].

Из видов голограмм можно выделить следующие:

радужная голограмма – это голограмма, позволяющая видеть 3D изображение при обычном белом дневном свете. При радужной голограмме на поверхности изображения получается большое количество линий, в одном миллиметре их количество может достигать десятка тысяч линий. Расстояние между этими линиями составляет значение длины волны света, которое фиксируется на изображении.

2D голограмма – представляет собой так же 3-х мерное изображение на некоторой поверхности, но в 2D голограммах основным способом сохранения информации являются диффракционные решётки, различающиеся частотой и углом наклона. На рисунке 1 можно видеть пример 2D голограммы [2].



Рисунок 1 - 2D голограмма

Этот вид диаграмм не совсем является голограммой, так как являются отдельными дифракционными элементами. Результат такой голограммы позволяет выводить 3-х мерные объекты разного цвета, кроме того, цвет каждого объекта будет изменяться в зависимости от угла, под которым на него смотрит наблюдатель.

2D/3D голограмма – это голограмма, имеющая в себе 2 изображения, где первое изображение представляет собой главный объект, а второе представляет собой фон этого изображения, который так же имеет трёхмерное представление. При повороте этого изображения относительно наблюдателя, так же меняется цвет и форма наблюдаемого объекта.

Пример 2D/3D голограммы можно видеть на рисунке 2 [2].



Рисунок 2 – Пример 2D/3D голограммы

Данный вид голограмм очень требователен к источнику освещения, и для более чёткого изображения ему необходимо более точечное освещение. В противном случае, если источник освещения не является подходящим, и выполнен, например, из обычных светильников, задний фон голограммы будет не чётким и размытым.

Следующим видом голограмм является 3D голограммы. 3D Голограммы воспроизводят 3D модель объекта, в полной их красочности и объёмности.

При повороте 3D голограммы по вертикали видно изменение цвета объекта, с отсутствием параллакса.

Такая голограмма наиболее часто используемая, и чаще она применяется в рекламных и декоративных целях. Так же по причине сложности копирования 3D голограммы, их могут использовать с целью защиты каких-либо документов.

Одним из ярких и известных примеров 3D голограммы можно привести голубя, изображённого на карте VISA. Его пример приведён на рисунке 3 [2].



Рисунок 3 - Пример 3D голограммы

Также стоит упомянуть и про более объёмные виды голограмм, которые представляют собой объёмную 3D фигуру, которую вполне можно обойти и рассмотреть со всех сторон.

В таких видах голограмм изображение формируется не на какой-либо поверхности, а появляется в пространстве, путём свечения нескольких лазеров, что в итоге даёт полноценное 3D изображение [2].

На рисунке 4 можно видеть пример такой объёмной голографии [4].



Рисунок 4 - Пример объёмной голографии

Что касается использования голографии в современном мире, то оно получило большое распространение в области реклам, шоу-бизнеса, обучения и даже медицины, где они используются для точечного изучения врачом малоизвестных ему операций, что сократит возможность врачебной ошибки.

Так же голографические устройства имеют большое применение в области голографических дисплеев, которые используются в самолётах, автомобилях, и многих других сферах. Такие дисплеи так же используются и в различных очках.

Такие дисплеи позволяют добавить к настоящему изображению, которое видно через стекло автомобиля или очков, с добавлением какого-либо трёхмерного изображения. Такие устройства вполне можно назвать устройствами дополнительной реальности [3].

1.2 Принципы работы голографии

Прежде чем изучать способы формирования голограмм., необходимо изучить что такое свет что он из себя представляет.

Свет в некоторой степени представляет собой волну.

При фотографировании на обычный фотоаппарат, он запоминает лишь амплитуду волны света, но в голограмме вместе с амплитудой, запоминается ещё и фаза волны, что позволяет сделать изображение более объёмным, путём фиксирования большего количества информации об объекте.

В основе работы голографических изображений лежат 2 процесса: дифракция и интерференция.

Дифракция – это явление, при котором световая волна, врезаясь в препятствие изменяет своё прямолинейного распространение [5].

Интерференция – это явление, в результате которого, при нахождении 2 волн друг на друга, формируется новая волна, амплитуда которой зависит от амплитуд столкнувшихся волн [6].

Процесс формирования голограмм связан с наложением друг на друга 2 волн, опорной и объектной, которые формируются в результате разделения лазерного луча. Опорной волной является световая волна, которая исходит от источника освещения, а объектной волной – волна, которая отражается от объекта, который записывается на голографическое изображение.

Важной частью системы записи голограммы является источник освещения. При начале разработки и создания голограмм у исследователей были проблемы именно из-за плохого источника освещения, но с появлением лазера, он стал наилучшим вариантом источника освещения в голографических системах.

Источник освещения необходим более точечный по той причине, что голографическое изображение является не просто результатом световой волны, а результатом наложения друг на друга 2 световых волн, а поэтому, требуется точность начально испускаемых волн.

Пример системы, которая выполняет съёмку голографического изображения представлен на рисунке 4 [4].

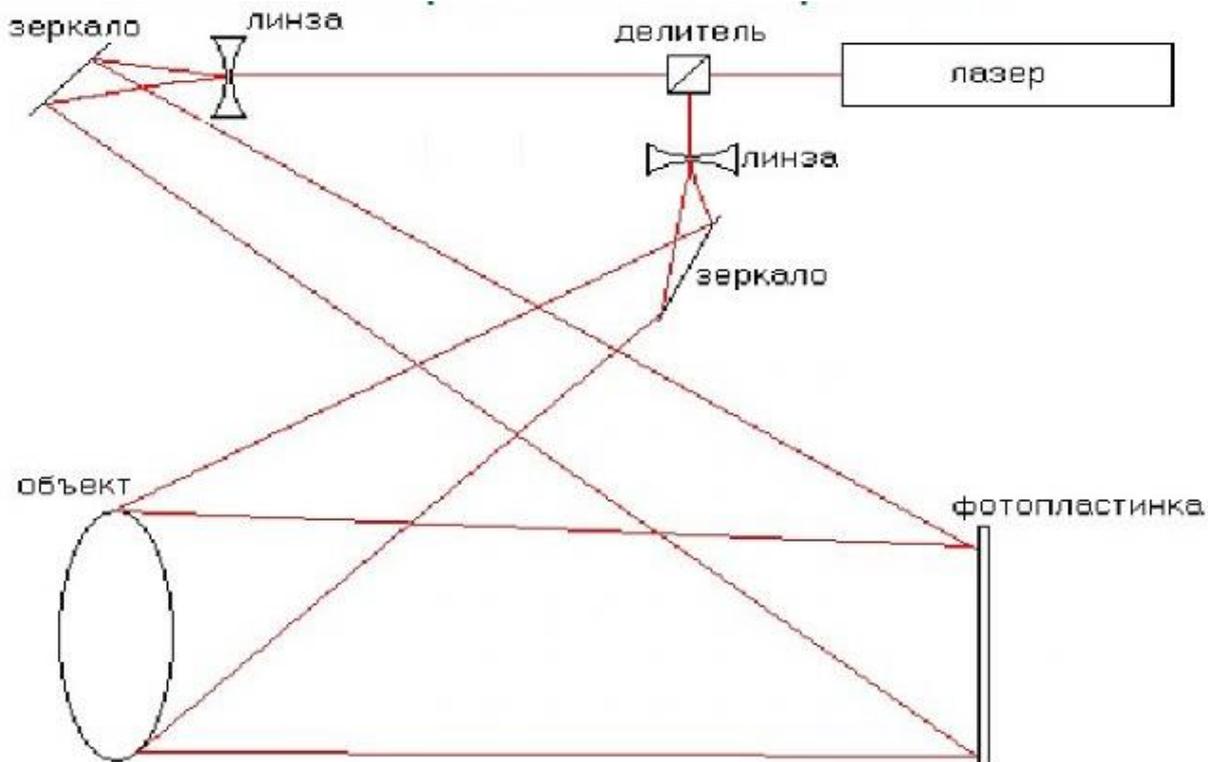


Рисунок 5 - Система считывания голографического изображения

Как можно видеть на рисунке 5, из источника освещения, который на схеме показан на лазер, исходит основное освещение, которое дальше делится на 2 волны, и далее через линзы подаются на фотопластину и объект.

В результате взаимодействия волны освещения с объектом который фоткается, происходит явление дифракции, которое изменяет получившуюся световую волну в зависимости от объекта.

Затем измененная объектная волна двигается в сторону фотопластины, где встречается с опорной волной, и в результате явления интерференции волна сохраняется на фотопластине.

После всех выполненных действий на фотопластине останется большое количество чёрно белых линий.

Для того что бы увидеть получившееся изображение необходимо просветить фотопластину световой волной, близкой к опорной волне.

В результате работы такой системы получается 3D изображение, отображённое на двухмерной поверхности. Рассмотренный вид голограмм называется радужным [4].

Для работы объёмных 3D изображений необходимо наличие специального устройства. Способного с помощью лазеров воспроизводить трёхмерное изображение. Такое устройство называется проектором.

Для работы таких проекторов, ему необходимо наличие некоторой прозрачной поверхности, чаще в их качестве используется прозрачная плёнка, а в более редких случаях могут быть использован и дым.

1.3 Анализ рынка голографических часов

Для начала будет рассмотрен продукт под названием UTP-80 от компании Uniel, который представляет собой часы с установленным на них проектором, который позволяет выводить проекцию времени и других данных на поверхности стены или потолка.

Корпус устройства представляет собой обычные часы, с добавленным с боку проекторов, который может крутиться вокруг своей оси на 180 градусов, что позволяет наводить его на стены или потолок.

В функционал этих часов также входит и возможности будильника.

Кроме времени, которое может выводиться как в 12-ти, так и в 24-х часовом режиме, часы могут выводить на экран и проекцию температуру помещения, в котором они находятся.

Питаться устройство может как от сети, так и от 3 батареек формата AAA. Между этими вариантами питания имеется лишь одно отличие, при питании от сети по кабелю USB, проектор будет работать до тех пор, пока его не выключит пользователь, в случае питания от батареек, проектор будет работать лишь 5 секунд, после чего автоматически выключится, что позволит экономить электроэнергию.

На рисунке 6 можно видеть пример рассматриваемых часов [7].



Рисунок 6 - Часы UTP-80

Корпус выполнен из чёрного пластика, и имеет размеры 135 x 94 x 33 мм. Цена на такое устройство составляет примерно 2400 рублей [7].

Следующим для рассмотрения был выбран голографический дисплей, позволяющий выводить изображение на основе быстрого кручения светодиодной ленты вокруг своей оси. Такие дисплеи предназначены для вывода любой желаемой информации с присутствием объёмного 3D изображение.

На рисунке 7 можно видеть пример рассматриваемого устройства [8].



Рисунок 7 - Голографический дисплей

Корпус такого устройства выполнен из пластика. С размеров 420x20x60 мм. На длинной грани корпуса расположена светодиодная лента, которая и воспроизводит изображение.

Для управления данным устройством имеется возможность подключения к нему по Wi-fi. Это позволяет выводить изображения большого количества форматов.

Для питания устройству необходимо 12 вольт, с максимальным током 3 Ампера.

Данная технология позволяет выводить 3D изображения над поверхностью крутящихся лопастей, что можно видеть на рисунке 8 [8].



Рисунок 8 - Пример использования рассматриваемого устройства

Цена на такое устройство составляет примерно 11 тысяч рублей и имеется в свободной продаже [8].

Также были найдены часы с подсвечиваемым прозрачным экраном, который позволяет рисовать на нём или оставлять надписи. От свечения прозрачного дисплея на часах, написанному тексту придаётся объёмная 3D форма.

Сами же часы выводят время на стандартном экране, переключая режимы отображения время, даты и дня недели. Так же имеется режим будильника.

На рисунке 9 можно видеть изображение таких часов.



Рисунок 9 - Часы с объёмным дисплеем для рисования

Эти часы поддерживают возможность работы как от сети, так и от батареек, при этом они имеют возможность заряжать 4 других устройств по USB, таких как телефон. Но эта возможность работает лишь при питании от сети.

Как и в других рассмотренных часах, при питании от батареек, часы не смогут светиться всегда, а лишь в течении 5 секунд после нажатия на их включение.

Корпус этого устройства выполнен из пластика и имеет размеры 138 x 117 x 64 мм.

Для рисования на дисплее используется специальный маркер, который идёт в комплекте для того, чтобы, его стереть, необходимо воспользоваться влажной салфеткой.

Цена на эти часы составляет примерно 1200 тысяч рублей [9].

Последним рассмотренным устройством с поддержкой голографии будут Зеркальные настольные часы с радио, с проектором на потолок Best Time, которые представляют собой аналог уже рассмотренных часов с поддержкой голографии, но являются более развитыми.

К основным отличиям между UTP-80 и Best Time можно отнести наличие 3 будильников, когда у UTP-80 их только 2, так же выводимое изображение на проекторе является более качественным и чётким.

Эти часы могут выводить время, температуру, влажность воздуха и дату. Кроме того, данные часы имеют возможность установки секундомера, что так же отсутствует на ранее рассмотренном устройстве.

Ещё одной дополнительной функцией является возможность настройки радиоприёмника, что превращает эти часы в мини-радио приёмник.

Для точной и качественной работы проектора на устройстве установлен датчик освещения, который позволит регулировать яркость работы проектора автоматически.

Питание устройства также похоже на то, что было реализовано в часах UTP-80. Но единственным отличием является то, что в часах Best Time, необходима постоянная батарейка формата CR2032, для сохранения настроек устройства в обесточенном состоянии. На рисунке 10 можно видеть внешний вид рассматриваемого устройства [10].

Так же на задней грани устройства установлен разъём USB, который позволит запитать какие-либо маломощное устройство., как например телефон.



Рисунок 10 - Часы с проектором Best Time

Корпус устройства имеет размер 185x75x55 и выполнен из пластика. Особенностью корпуса является то, что в нерабочем состоянии, экран часов представляет собой зеркальный дисплей.

Цена на это устройство составляет 4990 рублей [10].

1.4 Анализ требований к разрабатываемому устройству

После анализа уже готовых решений, имеющих на рынке, были выдвинуты требования к разрабатываемому устройству.

Устройство голографических часов должно иметь следующие функциональные блоки:

- блок вывода изображения – данный блок представляет собой несколько светодиодов, которые будут представлять собой пиксели выводимого изображения по вертикали.

- блок движения – это функциональный блок, который приводит в движение светодиоды для того, чтобы, создать динамическое изображение, которое в последствии и будет формировать голографию.

- блок управления – это блок, который управляет всеми элементами и блоками устройства.

К функционалу разрабатываемого устройства можно отнести следующее:

- Вывод голографического изображения настоящего времени;
- Настройка времени устройства.

Питание устройства должно осуществляться от сети, через кабель USB.

Корпус необходимо выполнить из пластика. По причине того, что, в устройстве будут использоваться механические движения большой частоты, необходимо предусмотреть защиту для пользователя, выполненную в виде пластиковой прозрачной оболочки.

Для эксплуатации устройства необходима температура окружающей среды в диапазоне от 5 до 25 градусов Цельсия, с относительной влажностью воздуха, не превышающей 80 процентов, и не ниже 40 процентов. Для

эксплуатации также необходимо обеспечить что бы работы устройства производилась без угрозы

Для хранения устройства необходима температура окружающей среды от 0 до 35 градусов Цельсия, с относительной влажностью воздуха от 30 до 80 процентов. При этом, необходимо обеспечение хранения без возможности механического повреждения корпуса устройства.

2 КОНСТРУКТОРСКИЙ РАЗДЕЛ

2.1 Структурная схема разрабатываемого устройства

Для начала разработки устройства необходимо выделить в системе структурные сегменты, их функции и взаимосвязь между собой. После данного этапа будет создана структурная схема устройства.

Для разрабатываемого устройства нужно выделить следующие структурные блоки: блок питания, генератор, вращающуюся часть и вывод изображения.

Блок, отвечающий за питание системы, представляет собой любой источник питания со схемой питания.

В блок «стационарная» входит генератор. Этот блок будет подключён к блоку питания, тем самым он будет приводить все остальные блоки в действие.

Задачей блока «вращающейся части» является прием и обработка данных о времени, благодаря которому, с помощью светодиодов будет выводиться изображение, то есть время. Для получения данных о времени в нём установлен датчик времени, который считывает реальное время, и на основе этих данных вращающаяся часть с помощью микроконтроллера будет обрабатывать данные и раскручиваться в одном направлении с контролируемой скоростью для вывода.

Связь между стационарной частью и вращающейся частью является однонаправленной, и направлена от блока «генератор», к блоку вращения. Обратная связь в этом случае не имеет смысла быть.

Связи с блоком, отвечающим за питание, будет установлена только с генератором, по причине необходимости балансировки питания.

Блок, отвечающий за вывод изображения, может иметь связь только с блоком вращающейся части. В этом блоке установлены только светодиоды.

На рисунке 11 изображена полученная структурная схема разрабатываемого устройства.



Рисунок 11 - Структурная схема устройства

На рисунке 11 можно видеть направление потока данных и сигналов, обозначенное стрелочкой. Передача сигналов осуществляется между блока генератора и блока вращающейся части, далее передаются блоку вывода изображения, как это и было описано выше.

2.2 Выбор комплектующих для разрабатываемого устройства

Следующим шагом необходимо выбрать модели используемых в проекте элементов и комплектующих.

Микроконтроллер управляющий системой освещения и эффектами освещения в проектируемом устройстве обладает требованиями по присутствию АЦП, необходим минимум 1 канал встроенного АЦП, количеству цифровых линий ввода-вывода, которых должно быть минимум 2, объёму флэш-памяти и питания микроконтроллера, которое должно

составлять не более 3.3 В. Также важным фактором при выборе микроконтроллера будет являться его цена.

Для сравнения были выбраны несколько наиболее подходящих микроконтроллеров ATmega328p, ATmega168PA, LGT8F328P, STM8S003K3T6C, из которых будет выбран один наиболее подходящий микроконтроллер.

Что касается питания, то по этому требованию подходят лишь микроконтроллеры ATmega и LogicGreen (LG).

Объём флэш-памяти, предназначенной для хранения программного кода, у каждого микроконтроллера разный, и имеет значение с 512 байт до 16 кбайт, у разных микроконтроллеров. Более подробно это будет рассмотрено в сравнительной таблице.

Количество контактов и размеры у этих микроконтроллеров существенно различается. Так если, у микроконтроллера STM8S003K3T6C имеется 28 линий ввода-вывода, то у ATmega168PA их количество меньше, и составляет всего 4 линии ввода-вывода. Микроконтроллер ATmega328p имеет 22 линии ввода-вывода, а LGT8F328P – 24.

В таблице 1 можно видеть сравнительные характеристики вышеперечисленных микроконтроллеров [11].

Таблица 1 - Сравнение микроконтроллеров

	ATmega328p	ATmega168PA	LGT8F328P	STM8S003K3T6C
Цена	400	350	300	150
Линий ввода-вывода	22	22	24	28
Частота, МГц	0-16	0-16	0-32	0-16
Объём флэш-памяти, кбайт	32	16	32	8
Питание, В	3.35-12	4.5-12	4-12	2.95-5.5
Длина, мм	33	33	43.6	7
Ширина, мм	18	18	18.3	7

После выполнения сравнения параметров микроконтроллеров, был выбран микроконтроллер ATmega328p.

Питание этого микроконтроллера подходит под указанные требования. Имеется 32кб флеш-памяти, подходящие размеры. Количество каналов вполне подходит под требования проекта.

Объём флэш-памяти отстаёт от объёма флэш-памяти других микроконтроллеров, но стоимость выбранного микроконтроллера является более низкой. Считается, что объёма флэш-памяти в размере 1 Кбайта будет достаточно. Кроме того, количество линий ввода-вывода не на много превышает требуемых, так схема не будет сильно загромождена [12].

Так основными причинами выбора микроконтроллера ATmega328p являются приемлемая стоимость, подходящее питание, подходящее количество контактов и флэш-памяти.

Следующим будет выбран вентилятор для задачи генератора, осуществляющий вращение.

Задачами этого устройства является вращение устройства, а потому, требования, предъявляемые к вентилятору, будут связаны с параметрами скорости, размерами и уровню шума.

Для вентилятора выдвигаются требования количеству оборотов и размерам, чем больше количество оборотов, тем шире табло, и питание должно составлять не более 15 В. Для сравнения между собой были выбраны следующие модели: FB5010S-12L3, DX120T, ACF3-FC00110.11.

Рассмотрим модель DX120. Его максимальное количество оборотов составляет 2200 об/м, уровень шума равен 36 дБ, а размер 120 x 120мм. Количество оборотов, уровень шума и размер удовлетворяет выдвинутым требованиям.

Модель FB5010S-12L3. Его максимальное количество оборотов составляет 4200 об/м, уровень шума равен 24 дБ, а размер 50 x 50мм. Такое количество оборотов и уровень шума удовлетворяет выдвинутым требованиям, однако, такой размер нам не подойдет.

Последняя рассматриваемая модель ACF3-FC00110.11. Его максимальный уровень шума равен 24 дБ, максимальное количество оборотов составляет 1000 об/м, а размер 120 x 120 мм. Уровень шума и размер удовлетворяют требованиям, но количество оборотов будет маленьким.

Сравнение этих моделей также представлено в таблице 2 [13].

Таблица 2 - Сравнение моделей вентиляторов

	FB5010S-12L3	DX120T	ACF3-FC00110.11
Цена, р	100	300	400
Максимальный уровень шума, дБ	24	36	24
Количество оборотов, об/м	4200	2200	1000
Размер, мм	50 x 50	120 x 120	120 x 120

По результатам сравнения моделей вентиляторов была выбрана модель DX120T по причине подходящего приемлемого количества оборотов, цены, размера и уровня шума.

Так же необходимо выбрать светодиодную ленту, предназначенную для световых эффектов. Для сравнения выбраны ленты на чипе WS2811, WS2812, WS2813.

В проекте будут использоваться именно светодиоды, а не светодиодная лента, так как в этом случае имеется возможность управления каждым светодиодом, что позволяет вывести нужное нам изображение.

2.3 Разработка программы и устройства

На основании выбранных элементов устройства, в этом разделе будет выполнена разработка принципиальной схемы, а в последствии и печатной платы устройства, после чего написан программный код

Для начала будет разработана принципиальная схема устройства, на которой обозначаются все элементы схемы, их контакты и обозначения, так

же на принципиальной схеме выстраивается связь между контактами элементов схемы.

При проектировании принципиальной схемы необходимо выполнить несколько основных подсистем.

На рисунке 12 можно видеть полученную принципиальную схему, которая в последствии будет детально рассмотрена. В приложении В можно видеть принципиальную схему в большем масштабе.

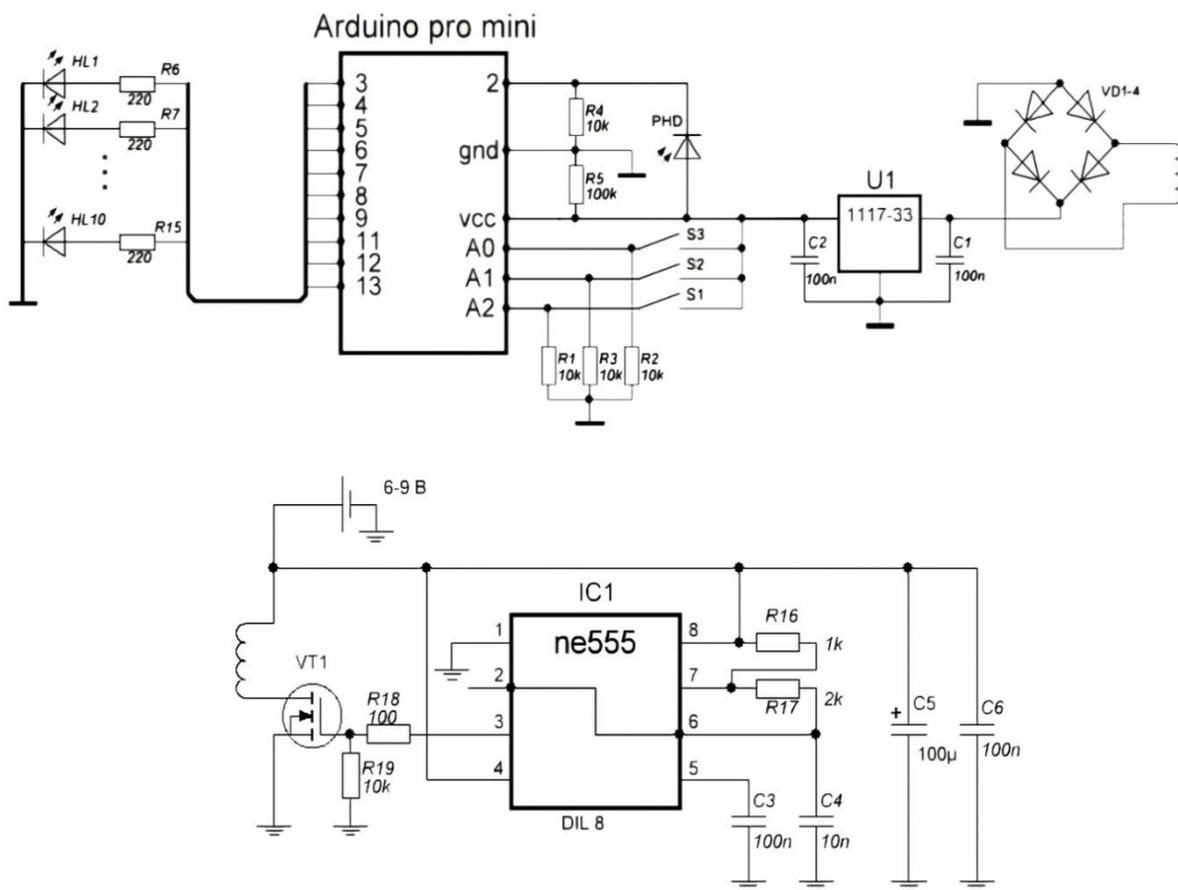


Рисунок 12 - Принципиальная схема устройства

Основной источник питания системы обычный зарядник телефона и имеет номинальное напряжение 9 В. Собранная схема питания позволяет производить безопасное питание, с возможностью его отключения от сети.

В основе схемы питания системы лежат модуль программируемого таймера и осциллятора NE555, предназначенный для включения/выключения питания.

Схема отвечающая за питание и раскручивание системы показана на рисунке 13.

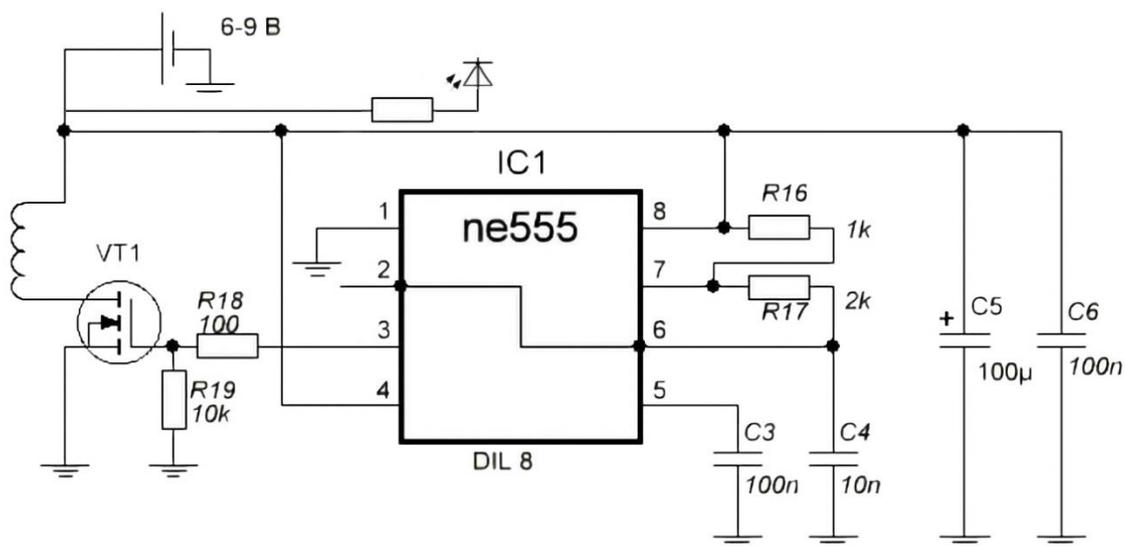


Рисунок 13 – Принципиальная схема генератора

Генератор основан на микросхеме-таймере NE555 и рассчитан на частоту около 26 кГц. Полевой транзистор - любой с допустимым током от 2А и напряжением от 30В. В данном варианте использовался полевой транзистор с компьютерной материнской платы. Диапазон питающих напряжений генератора лежит в пределах 5-15 В. Настройка производится в последнюю очередь подбором питающего минимального питающего напряжения, достаточного для нормальной работы схемы на приемной части. Дополнительно к источнику питания генератора нужно, через мягкие провода, подключить инфракрасный светодиод, а также необходимо рассчитать гасящий резистор для ограничения его тока.

Схема вращающейся части представляет собой простой диодный мост из быстродействующих диодов обозначенными VD1-4 и стабилизатором напряжения обозначенным "1117-33" в корпусе SOT-223 на 3.3 В). Диоды подойдут серии "FR" или "HER". В этом варианте стоят FR102. Стабилизатор - любой подходящий по расположению выводов и током от 500мА.

Принципиальная схема вращающейся части показана на рисунке 14.

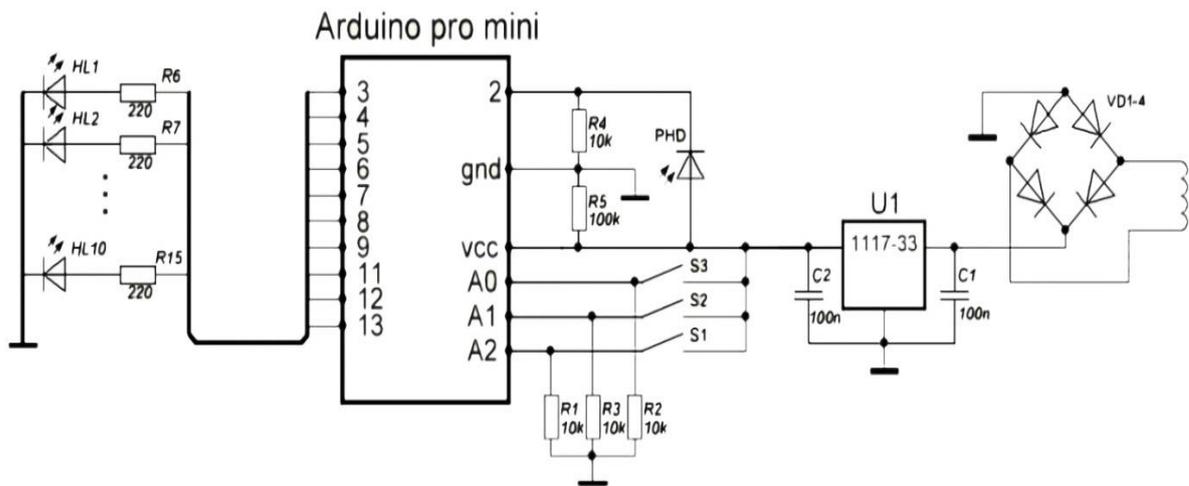


Рисунок 14 – Принципиальная схема вращающейся части

Для сборки данной системы нам понадобится печатная плата, которая представлена на рисунке 15.

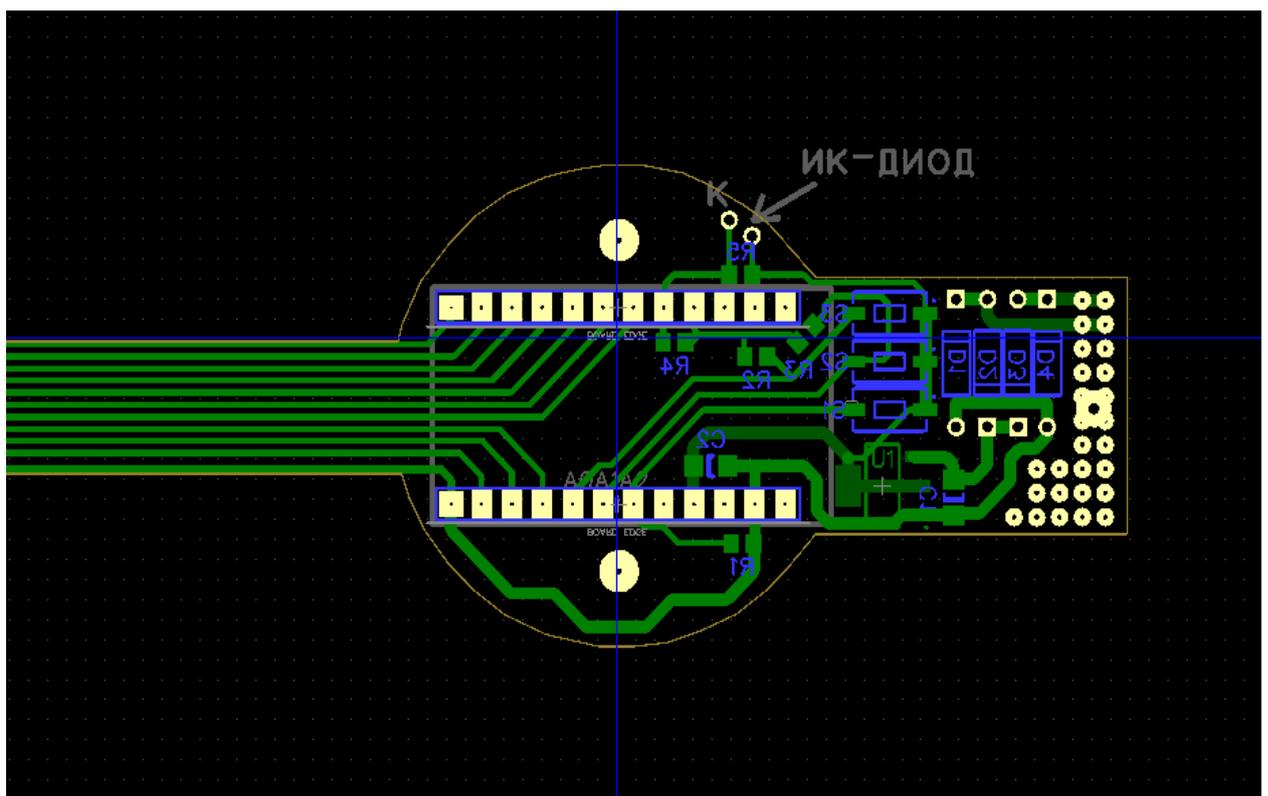


Рисунок 15 – Печатная плата

Все элементы подписаны. Отверстия справа не нуждаются в сверлении — это места для установки противовеса во время балансировки, то есть устранения вибраций при вращении. В данном случае противовес - винт

M3x20, установленный вертикально. Балансировку лучше производить после того, как убедились в работе часов. Иначе при исправлениях, возможно, потребуется очередная балансировка.

Далее нам нужно распечатать плату индикатора, которая представлена на рисунке 16.

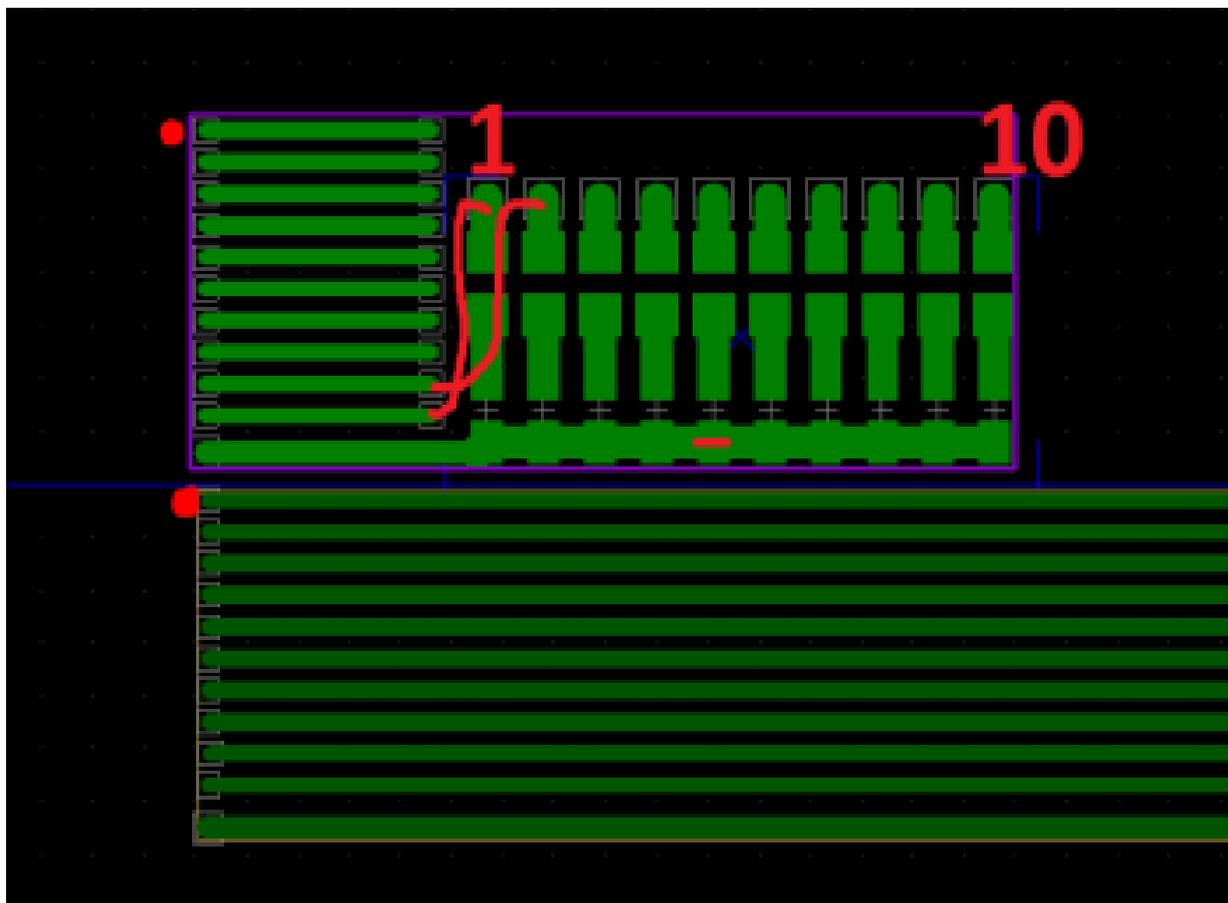


Рисунок 16 – Печатная плата индикатора

Печатная плата "индикатора" не имеет связи с основной, т.к. у меня не получались очень маленькие дорожки. Поэтому каждый светодиод соединен при помощи тонкого провода. Плата индикатора также подверглась переработке, пришлось заменить светодиоды 3мм на SMD 0805. Это позволило уменьшить размер табло и повысить четкость символов. Здесь указаны точками места совмещения, цифры - нумерация светодиодов, красными линиями - подключение (остальные светодиоды по порядку).

Платы надо печатать зеркально, если использовать технологию "ЛУТ".

Уже вытравленная, залуженная и оформленная плата крепится винтами М3 к ротору. Центры отверстий на плате имеются.

После пайки индикатора к плате, желательно усилить ее при помощи стропы из одножильного провода. Припаивать ее лучше к самой широкой дорожке на обеих платах - к минусу. В результате образуется треугольник, позволяющий не сорваться плате индикатора при больших оборотах. Желательно после проверки на предмет замыкания соседних дорожек залить термоклеем место стыка двух плат.

Схема с видом сверху представлена на рисунке 17.

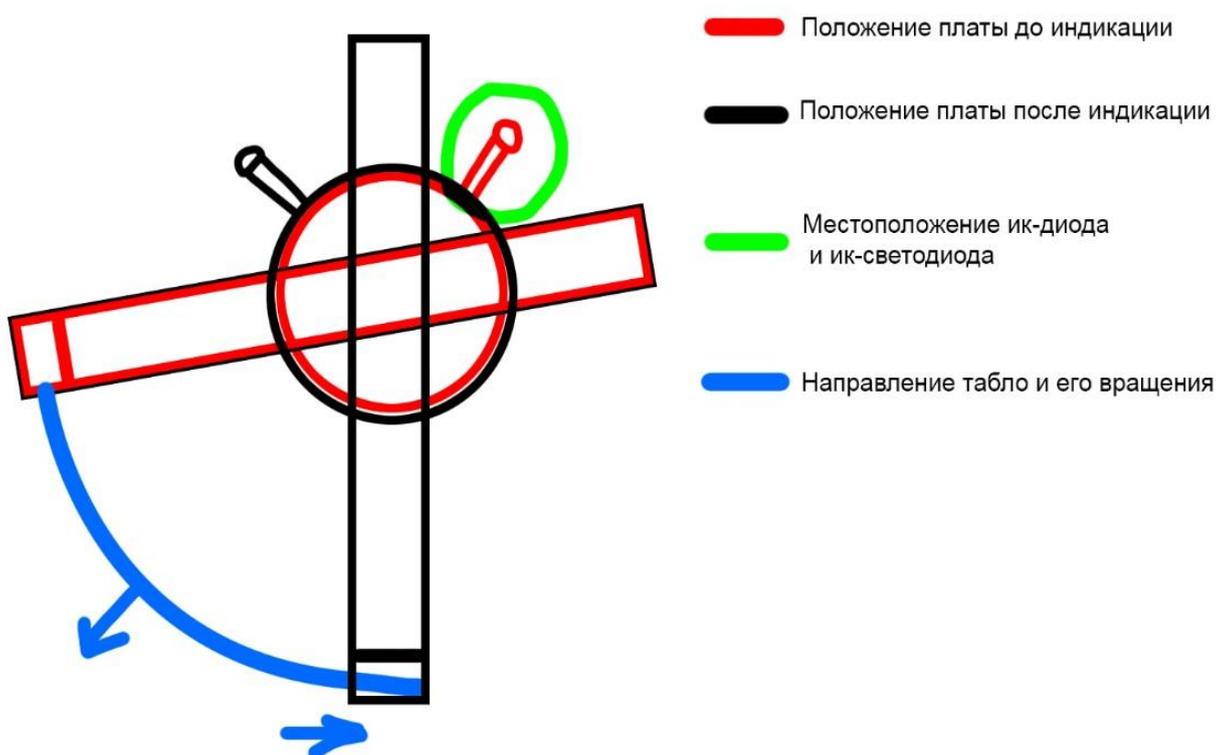


Рисунок 17 – Вид сверху

Далее разработаем корпус для устройства. Вид корпуса представлен на рисунке 18.

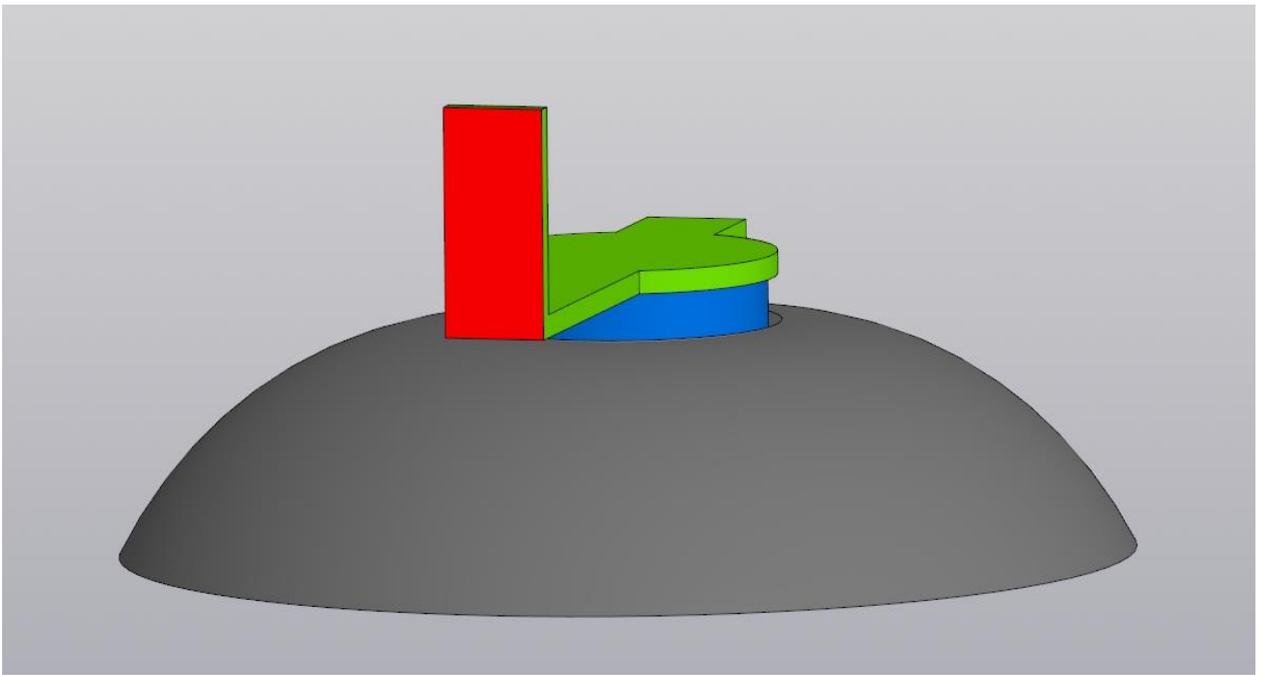


Рисунок 18 – Разработка корпуса

Теперь после завершённой разработки печатной платы устройства и корпуса, необходимо написать программный код, управляющий светодиодами.

В разрабатываемом программном коде необходимо расположить переменные, содержащие все необходимые настройки об устройстве, в которые входит контакт микроконтроллера, к которому подключены светодиоды, количество светодиодов, интервал входных значений от светодиода и так далее.

При запуске программы необходимо, что бы устройство было отключено, и лишь потом, при принятии данных от светодиодов, могла делать некоторую обработку и менять табло.

Кратко алгоритм работает следующим образом, имеется массив данных, имеющий длину равную количеству светодиодов. В этом массиве хранятся данные о времени.

На рисунке 19 можно видеть блок-схему алгоритма программы.

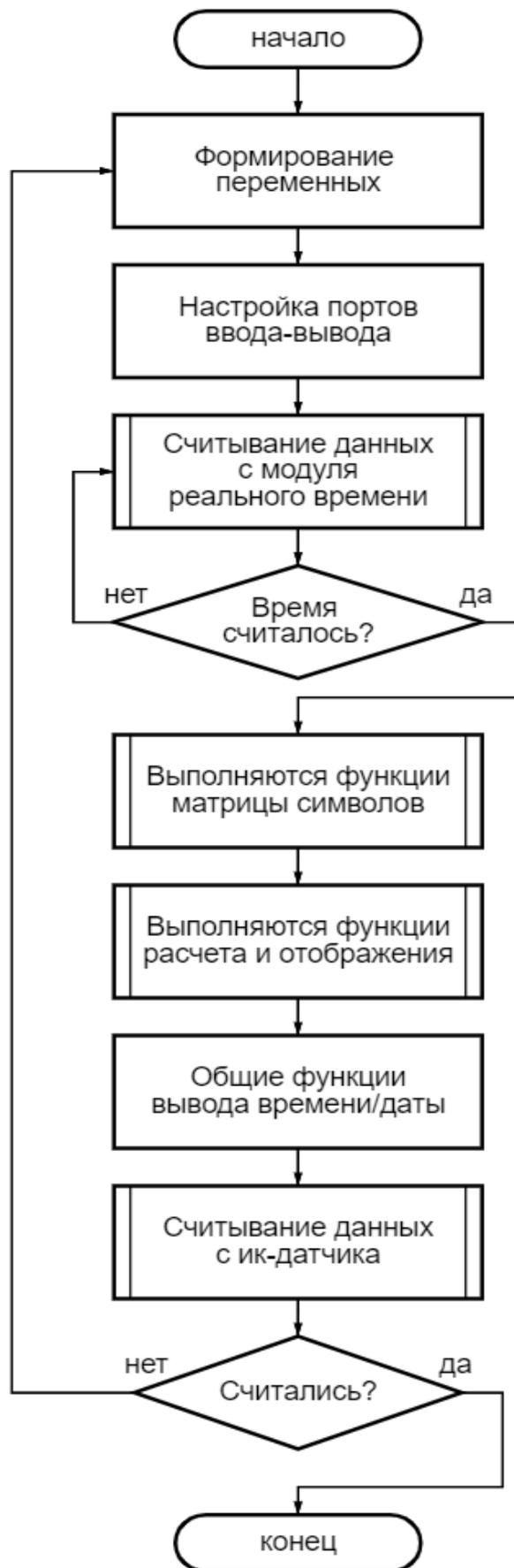


Рисунок 19 – Блок-схема алгоритма программы

Как можно видеть на рисунке 19, основное тело цикла работы программы состоит из бесконечного цикла, в котором происходит обработка данных, полученных с модуля с реального времени.

На блок-схеме, показанной на рисунке 19, можно видеть следующие функции:

Функция матрицы символов служит для преобразования данных в символы. Символы матрицы данных имеют прямоугольную, обычно квадратную форму и состоят из квадратных "ячеек", которые представляют биты.

Функция расчета и отображения служит для отображения этих самых символов.

Более подробная блок-схема по коду представлена на рисунке 20.

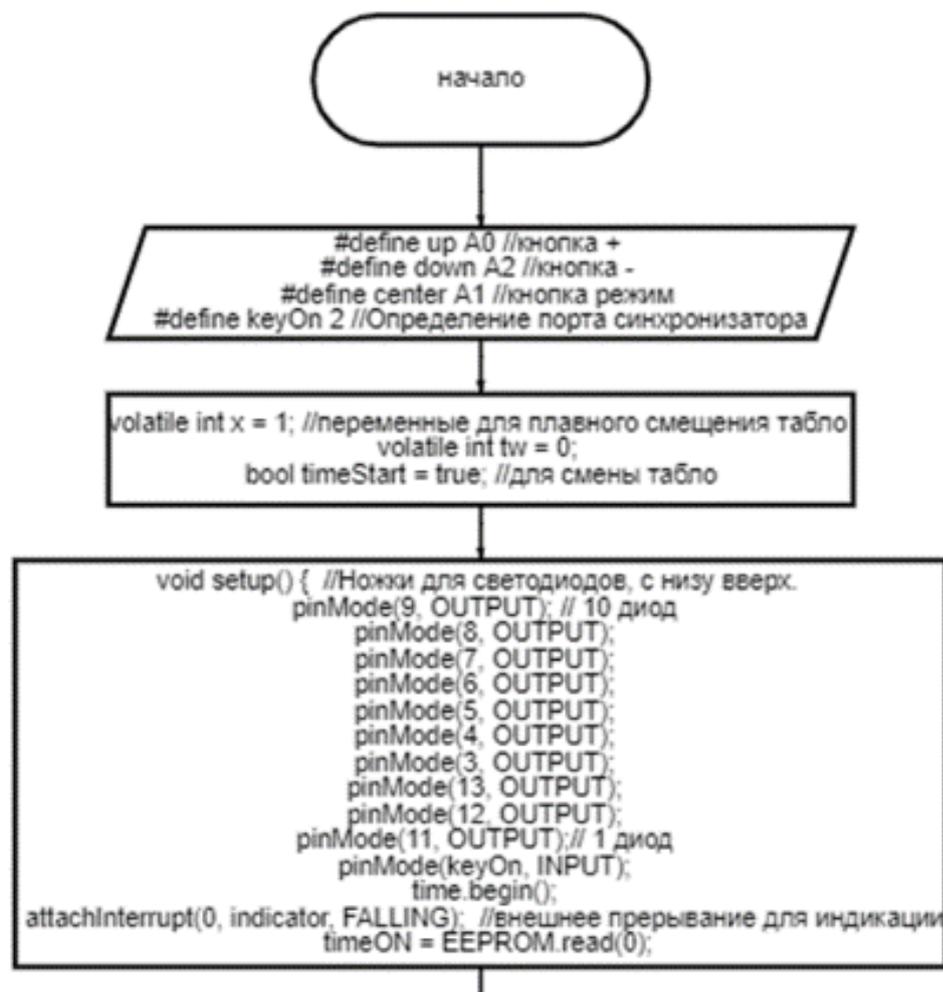


Рисунок 20 – Подробная блок-схема алгоритма программы

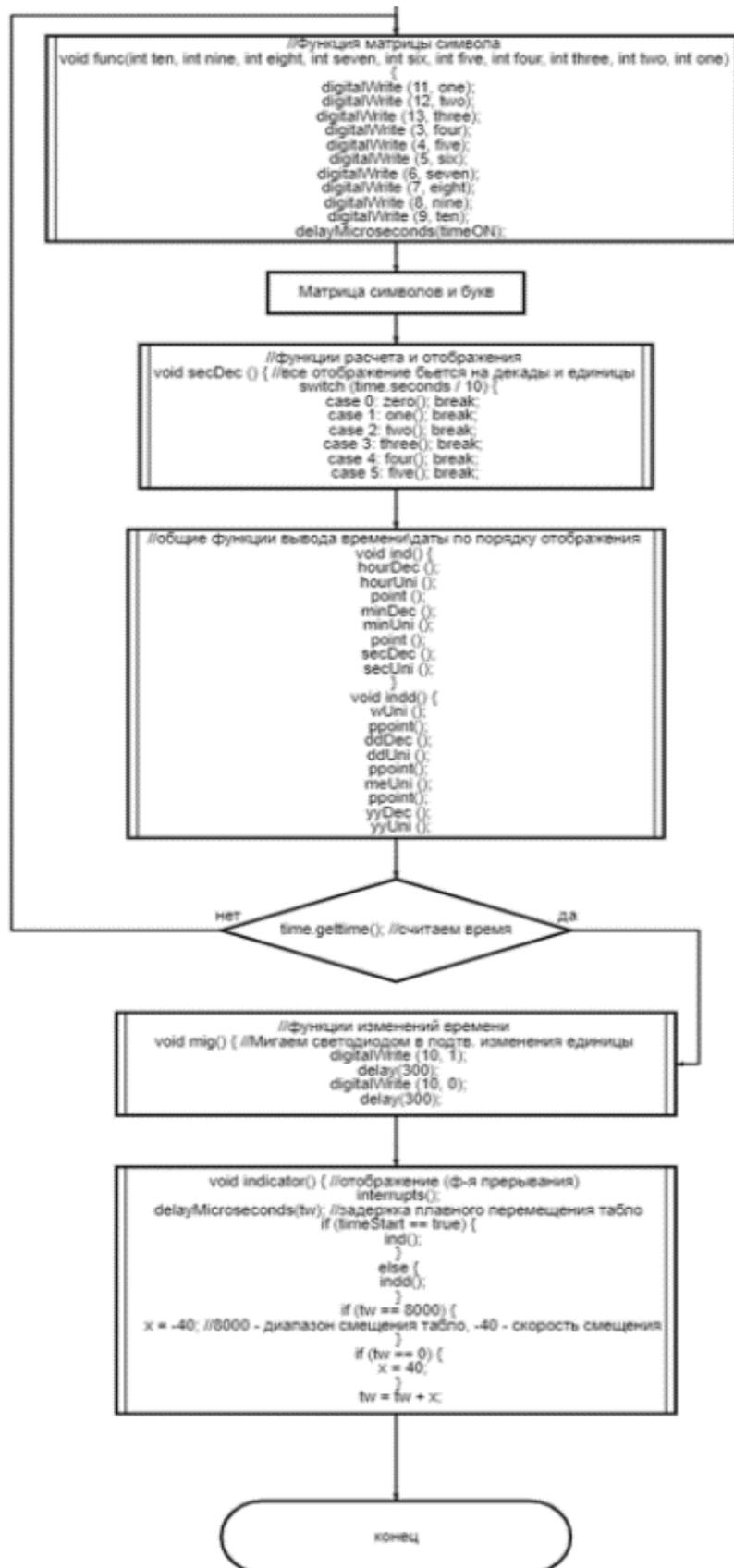


Рисунок 20 – Подробная блок-схема алгоритма программы

Полученный программный код представлен в приложении А.

3 ОХРАНА ТРУДА И ПРОМЫШЛЕННАЯ ЭКОЛОГИЯ

3.1 Анализ вредных и опасных факторов на рабочем месте веб-дизайнера

Опасные и вредные производственные факторы по природе возникновения делятся на следующие группы:

- физические;
- химические;
- психофизиологические;
- биологические.

В помещении на программиста могут негативно действовать следующие физические факторы:

- повышенная и пониженная температура воздуха;
- чрезмерная запыленность и загазованность воздуха;
- повышенная и пониженная влажность воздуха;
- недостаточная освещенность рабочего места;
- превышающий допустимые нормы шум;
- повышенный уровень ионизирующего излучения;
- повышенный уровень электромагнитных полей;
- повышенный уровень статического электричества;
- опасность поражения электрическим током;
- блеклость экрана дисплея.

К химически опасным факторам, постоянно действующим на веб-дизайнера, относятся активные частицы, возникающие в результате ионизации воздуха при работе компьютера.

Биологические вредные производственные факторы в данном помещении отсутствуют.

К психологически вредным факторам, воздействующим на программиста в течение его рабочей смены можно отнести следующие:

- нервно - эмоциональные перегрузки;
- умственное напряжение;
- перенапряжение зрительного анализатора.

Далее более подробно рассмотрены опасные и вредные факторы, воздействующие на программиста, возникшие в связи с разработкой данной системы.

3.1.1 Микроклимат рабочей зоны программиста

Микроклимат производственных помещений — это климат внутренней среды этих помещений, который определяется действующими на организм человека температурой, влажностью и скоростью движения воздуха

Кабинет программиста является помещением I категории (выполняются легкие физические работы), поэтому должны соблюдаться следующие требования:

Таблица 3 – Требования к микроклимату рабочей зоны

Период года	Категория работ	Температура воздуха, С	Относит. влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Легкая – 1	22-24	40-60	0,1
Холодный	Легкая – 1	21-23	40-60	0,1
Теплый	Легкая – 1	23-25	40-60	0,1
Теплый	Легкая – 1	22-24	40-60	0,2

Для создания и автоматического поддержания независимо от наружных условий оптимальных значений температуры, влажности, чистоты и скорости движения воздуха, в холодное время года используется водяное отопление, в теплое время года применяется кондиционирование воздуха. Кондиционер представляет собой вентиляционную установку, которая с

помощью приборов автоматического регулирования поддерживает в помещении заданные параметры воздушной среды.

Воздействие шума на программиста. Защита от шума.

Источниками шумовых помех могут стать вентиляционные установки, кондиционеры или периферийное оборудование для ЭВМ (плоттеры, принтеры и др). Длительное воздействие этих шумов отрицательно сказывается на эмоциональном состоянии персонала.

Согласно ГОСТ 12.1.003-76 ССБТ эквивалентный уровень звука не должен превышать 50 дБА. Для того, чтобы добиться этого уровня шума рекомендуется применять звукопоглощающее покрытие стен.

В качестве мер по снижению шума можно предложить следующее:

- облицовка потолка и стен звукопоглощающим материалом (снижает шум на 6-8 дб);
- экранирование рабочего места (постановкой перегородок, диафрагм);
- установка в компьютерных помещениях оборудования, производящего минимальный шум;
- рациональная планировка помещения.

Защиту от шума следует выполнять в соответствии с ГОСТ 12.1.003-76, а звукоизоляция ограждающих конструкций должна отвечать требованиям главы СНиП «Защита от шума. Нормы проектирования».

3.2 Освещение рабочего места

Работа, выполняемая с использованием вычислительной техники, имеют следующие недостатки:

- вероятность появления прямой блескости;
- ухудшенная контрастность между изображением и фоном;
- отражение экрана.

В связи с тем, что естественное освещение слабое, на рабочем месте должно применяться также искусственное освещение. Далее будет произведен расчет искусственного освещения.

Размещение светильников определяется следующими размерами:

$H = 3$ м. - высота помещения

$h_c = 0,25$ м. - расстояние светильников от перекрытия

$h_n = H - h_c = 3 - 0,25 = 2,75$ м. - высота светильников над полом

h_p = высота расчетной поверхности = 0,7 м (для помещений, связанных с работой ПЭВМ)

$h = h_n - h_p = 2,75 - 0,7 = 2,05$ - расчетная высота. (1)

Светильника типа ЛДР (2x40 Вт). Длина 1,24 м, ширина 0,27 м, высота 0,10 м.

L - расстояние между соседними светильниками (рядами люминесцентных светильников), L_a (по длине помещения) = 1,76 м, L_b (по ширине помещения) = 3 м.

l - расстояние от крайних светильников или рядов светильников до стены,

$l = 0,3 - 0,5L$. (2)

$l_a = 0,5L_a$, $l_b = 0,3L_b$

$l_a = 0,88$ м., $l_b = 0,73$ м.

Светильники с люминесцентными лампами в помещениях для работы рекомендуют устанавливать рядами.

Метод коэффициента использования светового потока предназначен для расчета общего равномерного освещения горизонтальных поверхностей при отсутствии крупных затемняющих предметов. Потребный поток ламп в каждом светильнике

$$\Phi = E \cdot r \cdot S \cdot z / N \cdot h \quad (3)$$

где E - заданная минимальная освещенность = 300 лк., т. к. разряд зрительных работ = 3

r - коэффициент запаса = 1,3 (для помещений, связанных с работой ПЭВМ)

S - освещаемая площадь = 30 м².

z - характеризует неравномерное освещение, $z = E_{ср} / E_{min}$ - зависит от отношения $l = L/h$, $la = La/h = 0,6$, $lv = Lv/h = 1,5$. Т. к. l превышают допустимых значений, то $z=1,1$ (для люминесцентных ламп).

N - число светильников, намечаемое до расчета. Первоначально намечается число рядов n , которое подставляется вместо N . Тогда Φ - поток ламп одного ряда.

$N = \Phi / \Phi_1$, где Φ_1 - поток ламп в каждом светильнике.

h - коэффициент использования. Для его нахождения выбирают индекс помещения i и предположительно оцениваются коэффициенты отражения поверхностей помещения $r_{пот.}$ (потолка) = 70%, $r_{ст.}$ (стены) = 50%, $r_{п.}$ (пола) = 30%.

$$\Phi = 300 \cdot 1,3 \cdot 25 \cdot 1,1 / 2 \cdot 0,3 = 21450 \text{ лм.}$$

Я предлагаю установить два светильника в ряд. Светильники вмещаются в ряд, так как длина ряда около 4 м. Применяем светильники с лампами 2x40 Вт с общим потоком 5700 лм. Схема расположения светильников представлена на рисунке 21.

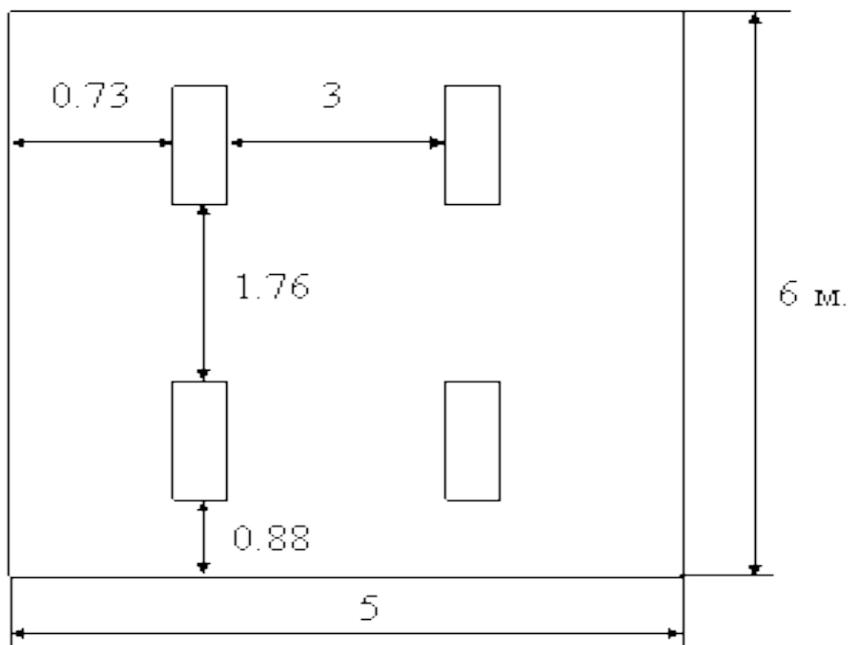


Рисунок – 21 Схема расположения светильников

3.3 Утилизация средств вычислительной техники и оргтехники

3.3.1 Этапы утилизации оргтехники

Первый этап производится силами организации с привлечением сертифицированных экспертов. Последующие этапы имеет право выполнять одна из специализированных фирм, зарегистрированных в Пробирной палате РФ.

1. Получение акта списания:

- 1) экспертная оценка;
- 2) определение остаточной стоимости списываемой техники;
- 3) заключение о невозможности ее дальнейшего использования;
- 4) оформление выбытия ОС.

2. Демонтаж оборудования с отделением содержащих драгоценные металлы частей:

1) обязателен акт об изъятии комплектующих, на основании которого техника списывается с карточек учета;

2) на его основании на каждую деталь заводится своя карточка учета (форма М-17), в которой указывается наименование лома (название узла), его общая масса и доля в ней драгметаллов каждого наименования.

3. Аффинаж – отделение драгоценных металлов от частей техники.

4. Оприходование полученных ценностей.

5. Передача драгметаллов в фонд государства.

3.2.2 Как провести утилизацию с помощью спец фирм

Организации, специализирующиеся на утилизации офисной и бытовой техники, имеют все необходимые разрешения и лицензии, а также оборудование для аффинажа драгоценных металлов. Многие из них уполномочены и проводить техническую экспертизу, предшествующую списанию такого оборудования. Специалисты осуществляют:

1) помощь в составлении актов списания на каждый из утилизируемых приборов;

- 2) транспортировку, обезвреживание и размещение опасной части отходов;
- 3) оформление санитарно-эпидемиологических документов;
- 4) все последующие процедуры, связанные с драгметаллами.

3.2.3 Алгоритм для представителей фирм-владельцев старой оргтехники

1. Выбрать компанию для утилизации техники, сертифицированную в Пробирной палате.
2. Составить предварительный список техники, подлежащей утилизации.
3. Согласовать с фирмой стоимость ее услуг в зависимости от количества единиц техники и предоставляемого сервиса.
4. Заключить договор на утилизацию списанных основных технических средств.
5. Демонтаж поименованной техники.
6. Вывоз демонтированной аппаратуры.
7. Подписание акта о выполнении работ и окончательный расчет.

3.4 Вывод по разделу

В разделе “Охрана труда и производственная экология” были рассмотрены следующие подразделы:

– “Выявление вредных и опасных факторов по отношению к веб-дизайнеру”, в котором были рассмотрены понятия вредного и опасного фактора, а также существующие вредные и опасные факторы на рабочем месте инженера-электронщика, и возможные последствия, с их разделением по классификации.

– “Обеспечение безопасности труда на рабочем месте "Веб-дизайнера"” в котором были рассмотрены правила проведения работ перед началом работы, во время работы и после выполнения работы, для

обеспечения безопасности работника и оборудования. Рассмотрены нормы времени работы работника, с обязательным отдыхом от работы.

– “Способы утилизации вычислительной техники и оргтехники” в котором были рассмотрены причины, по которым необходимо проводить утилизацию вычислительной и оргтехники, нормативные документы в соответствии с которыми проводится утилизация, а также некоторые этапы, утилизации вычислительной и оргтехники.

В соответствии с принятыми нормами в отделе информационных технологий обеспечивается необходимый микроклимат, минимальный уровень шума, созданы удобные и правильные с точки зрения эргономики рабочие места, соблюдены требования технической эстетики и требования к ЭВМ.

Для сотрудников отдела в процессе работы одним из важнейших факторов, влияющих на производительность труда при длительной зрительной работе, является достаточное освещение рабочего места. Это достигается правильным выбором и расположением осветительных приборов.

В целом условия труда веб-дизайнера соответствуют общепринятым нормам, сотрудникам обеспечены комфорт и благоприятные условия.

4 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКАЯ ЧАСТЬ

4.1 Расчет затрат на разработку программы для микроконтроллера

Затраты на разработку программы для микроконтроллера рассчитываются по формуле 4.

$$Z_{РПР} = Z_{ФОТР} + Z_{ОВФ} + Z_{ЭВМ} + Z_{СПП} + Z_{ХОН} + P_H, \quad (4)$$

где $Z_{ФОТР}$ – общий фонд оплаты труда разработчиков программы,

$Z_{ОВФ}$ – начисления на заработную плату разработчиков программы во внебюджетные фонды,

$Z_{ЭВМ}$ – затраты, связанные с эксплуатацией техники,

$Z_{СПП}$ – затраты на специальные программные продукты, необходимые для разработки программы для микроконтроллера,

$Z_{ХОН}$ – затраты на хозяйственно-операционные нужды (бумага, литература, носители информации и т.п.),

P_H – накладные расходы ($P_H = 30\%$ от $Z_{ФОТР}$).

При разработке программы для микроконтроллера общее время разработки составило 0,5 месяца.

Фонд оплаты труда за время работы над программой для микроконтроллера рассчитывается по формуле 5.

$$Z_{ФОТР} = \sum_{j=1}^m O_{Pj} \cdot T_{РПРj} \cdot (1 + k_D)(1 + k_Y), \quad (5)$$

где O_{Pj} – оклад j -го разработчика. В разработке участвовал 1 человек, его оклад составляет 25000 руб.,

$T_{РПРj}$ – общее время работы над программой в месяцах, $T_{РПР} = 0,5$,

k_D – коэффициент дополнительной зарплаты, $k_D = 20\% = 0,2$,

k_Y – районный коэффициент, $k_Y = 0,15$.

Таким образом,

$$Z_{ФОТР} = 25000 \cdot 0,5 \cdot (1 + 0,2) \cdot (1 + 0,15) = 17250 \text{ руб.}$$

Страховой взнос во внебюджетные фонды складываются из взносов на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний.

Страховые взносы складываются из обязательного пенсионного страхования (ОПС), отчислений в фонд социального страхования и отчислений в фонд обязательного медицинского страхования.

Значения всех используемых ставок приведены в таблице 4.

Таблица 4 – Значения ставок страховых взносов

№	Наименование внебюджетного фонда	Размер ставок, %
1	Пенсионный фонд	22
2	Фонд социального страхования	2,9
3	Федеральный фонд обязательного медицинского страхования	5,1
	Итого:	30

Сумма начислений на заработную плату во внебюджетные фонды рассчитывается по формуле 6.

$$Z_{\text{ОВФ}} = 0,3 \cdot Z_{\text{ФОТР}}, \quad (6)$$

$$Z_{\text{ОВФ}} = 0,3 \cdot 17250 = 5175 \text{ руб.}$$

Затраты, связанные с использованием вычислительной и оргтехники, рассчитываются по формуле 7.

$$Z_{\text{ЭВМ}} = T_{\text{МРПР}} \cdot k_{\Gamma} \cdot n \cdot C_{\text{М-ч}} \quad (7)$$

где k_{Γ} – коэффициент готовности ЭВМ, $k_{\Gamma} = 0,95$,

n – количество единиц техники, равно 1,

$C_{\text{М-ч}}$ – себестоимость машино–часа, $C_{\text{М-ч}} = 8$ руб.,

$T_{\text{МРПР}}$ – машинное время работы над программой, равно 0,5 мес.

Перевод рабочего времени в часы осуществляется по формуле 8.

$$T_{\text{час}} = T_{\text{мес}} \cdot Ч_{\text{РД}} \cdot T_{\text{см}} \cdot K_{\text{см}}, \quad (8)$$

где $T_{\text{час}}$ – рабочее время, ч,

$T_{\text{мес}}$ – рабочее время, мес, ($T_{\text{мес}} = 0,5$),

$Ч_{\text{рд}}$ – число рабочих дней, ($Ч_{\text{рд}} = 22$),

$T_{\text{см}}$ – продолжительность рабочей смены, ($T_{\text{см}} = 8$ ч),

$K_{\text{см}}$ – количество рабочих смен, ($K_{\text{см}} = 1$).

Таким образом, время на разработку программы для микроконтроллера с использованием ЭВМ составляет:

$$T_{\text{час}} = 0,5 \cdot 22 \cdot 8 \cdot 1 = 88 \text{ часа,}$$

$$З_{\text{ЭВМ}} = 88 \cdot 0,95 \cdot 1 \cdot 8 = 668,8 \text{ руб.}$$

Затраты на специальные программные продукты, необходимые для разработки программы для микроконтроллера рассчитываются по формуле 9.

$$З_{\text{СПП}} = \sum_{p=1}^n Ц_p \quad (9)$$

где $Ц_p$ – цена p -го специального программного продукта.

Перечень программных продуктов специального назначения приведен в таблице 2.

Таблица 5 – Программные продукты специального назначения

№	Название ПП	Цена, руб.
1	Среда разработки Arduino IDE	0
2	Proteus	0
	Итого:	0

Использованные программные продукты бесплатны, поэтому:

$$З_{\text{СПП}} = 0 \text{ руб.}$$

Затраты на хозяйственно–организационные равны нулю.

$$З_{\text{ХОИ}} = 0 \text{ руб.}$$

Накладные расходы рассчитываются по формуле 10.

$$P_H = З_{\text{ФОТР}} \cdot k_{\text{НР}}, \quad (10)$$

$$P_H = 17250 \cdot 0,3 = 5175 \text{ руб.}$$

Таким образом, затраты на разработку программы для микроконтроллера, рассчитанные по формуле 4, составят:

$$Z_{\text{РПР}} = 17250 + 5175 + 668,8 + 0 + 0 + 5175 = 28268,8 \text{ руб.}$$

4.2 Расчет затрат на внедрение программы для микроконтроллера

Затраты на внедрение программы для микроконтроллера (ЗВПР) рассчитываются по формуле 11.

$$Z_{\text{ВПР}} = Z_{\text{М}} + Z_{\text{КТС}} \cdot (1 + k_{\text{Тун}}) + Z_{\text{ПО}} + Z_{\text{ФОТВ}} + Z_{\text{ОФВ}} + Z_{\text{ЭВМ}} + P_{\text{КОМ}} + P_{\text{Н}}, \quad (11)$$

где $Z_{\text{М}}$ – затраты на приобретение материалов, руб.,

$Z_{\text{КТС}}$ – затраты на приобретение комплекса технических средств, руб.,

$Z_{\text{ПО}}$ – затраты на приобретение программного обеспечения (включают стоимость разработанной программы, а также других существующих ПП, необходимых для функционирования системы), руб.,

$Z_{\text{ФОТВ}}$ – затраты на оплату труда работников, занятых внедрением проекта, руб.,

$Z_{\text{ОФВ}}$ – страховой взнос во внебюджетные фонды с заработной платы работников, занятых внедрением проекта, руб.,

$Z_{\text{ЭВМ}}$ – затраты, связанные с эксплуатацией ЭВМ при внедрении проектного решения, руб.,

$P_{\text{КОМ}}$ – командировочные расходы, руб.,

$P_{\text{Н}}$ – накладные расходы, руб.,

$k_{\text{Тун}}$ – коэффициент транспортирования, установки и наладки комплекса технических средств, определяется действующими нормативами организации, а также спецификой конкретного проекта.

Затраты на приобретение материалов (ЗМ) приведены в таблице 6.

Таблица 6 – Затраты на приобретение материалов

Наименование	Цена за единицу (руб.)	Кол-во (шт.)	Всего (руб.)
Arduino mini	400	1	400
Программируемый таймер и осциллятор NE555	3	1	3

Продолжение таблицы 6

Линейный регулятор AMS1117 – 3,3	5	1	5
Часы реального времени DS3231	150	1	150
MOSFET-транзистор	30	1	30
Резистор 10кОм	2	5	10
Резистор 100кОм	2	1	2
Резистор 220Ом	2	10	20
Резистор 1кОм	2	1	2
Резистор 2кОм	2	1	2
Резистор 100	2	1	2
Конденсатор 100 нФ	10	2	20
Конденсатор 10 нФ	10	1	10
Конденсатор 100 мкФ	20	1	20
Выпрямительный диод FR102	5	4	20
Светодиод	1	10	10
Инфракрасный светодиод	3	1	3
Тактовая кнопка	1	3	3
		Итого	712

Дополнительного приобретения компьютеров или других КТС не требуется, следовательно, $Z_{\text{КТС}} = 0$.

Затраты на приобретение программного обеспечения в данном случае равны затратам на разработку и составляют $Z_{\text{ПО}} = 28268,8$ руб.

Внедрением занят один системный инженер с окладом 25000 руб. Время внедрения – 0,5 месяца. По формуле рассчитываем затраты на оплату труда и страховой взнос во внебюджетные фонды.

$$Z_{\text{ФОТВ}} = 25000 \cdot 0,5 \cdot (1 + 0,15) = 14375 \text{ руб.}$$

$$Z_{\text{ОВФ}} = 14375 \cdot 0,3 = 4312,5 \text{ руб.}$$

Затраты, связанные с эксплуатацией ЭВМ при внедрении проектного решения составят:

$$Z_{\text{ЭВМ}} = 0,5 \cdot 22 \cdot 8 \cdot 8 = 704 \text{ руб.}$$

Командировочные расходы при внедрении программы для микроконтроллера не планируются, следовательно, $R_{\text{ком}} = 0$.

Так как коэффициент накладных расходов по данным организации составляет $k_{НР} = 0,3$, то величина накладных расходов равна 4312,5 руб.

Суммарные затраты на внедрение составят:

$$З_{ВНР} = 712 + 0 + 28268,8 + 14375 + 4312,5 + 704 + 0 + 4312,5 = 52684,8 \text{ руб.}$$

3.3 Расчет эксплуатационных текущих затрат по программе для микроконтроллера

Годовые затраты на обработку результатов до внедрения разработанной программы для микроконтроллера рассчитываются по формуле 12.

$$C_1 = 3П_1 + ОТ_{ВН1} + З_{ЭВМ1} + M_{з1} + НР_1, \quad (12)$$

где $3П_1$ – затраты на оплату труда сотрудника на выполнение функций до внедрения проектного решения,

$ОТ_{ВН1}$ – страховой взнос во внебюджетные фонды,

$З_{ЭВМ1}$ – затраты, связанные с эксплуатацией ЭВМ,

$M_{з1}$ – годовые материальные затраты на сопровождение программы для микроконтроллера составляют 1000 руб.,

$НР_1$ – накладные расходы.

Временные затраты работы сотрудника в месяцах рассчитываются по формуле 13.

$$T_{1мес} = \frac{T_{1час}}{Ч_{рд} \cdot Ч_{рч}}, \quad (13)$$

где $T_{1мес}$, $T_{1час}$ – время, затрачиваемое сотрудником на обработку результатов, в месяцах и часах соответственно ($T_{1час} = 170$ часов),

$Ч_{рд}$ – число рабочих дней в месяц,

$Ч_{рч}$ – число рабочих часов в день.

$$T_{1мес} = \frac{170}{22 \cdot 8} = 1 \text{ мес.}$$

Тогда затраты на оплату труда сотрудника рассчитываются по формуле 14.

$$ЗП_1 = O_c \cdot T_{1мес} \cdot (1 + k_d) \cdot (1 + k_y), \quad (14)$$

где O_c – оклад сотрудника (оклад составляет 25000 руб.),

$$ЗП_1 = 25000 \cdot 1 \cdot (1 + 0,2) \cdot (1 + 0,15) = 34500 \text{ руб.}$$

Страховой взнос до внедрения вычисляют по формуле 15.

$$ОТ_{ВН1} = ЗП_1 \cdot 0,3, \quad (15)$$

$$ОТ_{ВН1} = 34500 \cdot 0,3 = 10350 \text{ руб.}$$

Рассчитываем затраты, связанные с эксплуатацией ЭВМ до внедрения по формуле 16.

$$З_{ЭВМ1} = T_{1час} \cdot C_{М-Ч}, \quad (16)$$

$$З_{ЭВМ1} = 170 \cdot 8 = 1360 \text{ руб.}$$

Подставив соответствующие значения в формулу 12, получим:

$$C_1 = 34500 + 10350 + 1360 + 1000 = 47210 \text{ руб.}$$

Годовые затраты на эксплуатацию системы после внедрения программы для микроконтроллера рассчитываются аналогично по формуле 17.

$$C_2 = ЗП_2 + ОТ_{ВН2} + З_{ЭВМ2} + М_{з2} + НР_2, \quad (17)$$

где $ЗП_2$ – затраты на оплату труда сотрудника после внедрения,

$ОТ_{ВН2}$ – страховой взнос во внебюджетные фонды,

$З_{ЭВМ2}$ – затраты, связанные с эксплуатацией ЭВМ после внедрения,

$М_{з2}$ – материальные затраты, годовые материальные затраты на сопровождение программы для микроконтроллера составляют 1500 руб.,

$НР_2$ – накладные расходы.

Временные затраты работы сотрудника в месяцах вычисляются по формуле 18.

$$T_{2\text{мес}} = \frac{T_{2\text{час}}}{\text{Ч}_{\text{рд}} \cdot \text{Ч}_{\text{рч}}}, \quad (18)$$

где $T_{2\text{мес}}$, $T_{2\text{час}}$ – время, затрачиваемое сотрудником на обработку результатов, в месяцах и часах соответственно ($T_{2\text{час}} = 70$ часов),

$\text{Ч}_{\text{рд}}$ – число рабочих дней в месяц,

$\text{Ч}_{\text{рч}}$ – число рабочих часов в день.

$$T_{2\text{мес}} = \frac{70}{22 \cdot 8} = 0,4 \text{ мес.}$$

Тогда затраты на оплату труда сотрудника высчитываются по формуле 19.

$$ЗП_2 = O_c \cdot T_{2\text{мес}} \cdot (1 + k_d) \cdot (1 + k_y), \quad (19)$$

где O_c – оклад сотрудника (оклад составляет 25000 руб.)

$$ЗП_2 = 25000 \cdot 0,4 \cdot (1 + 0,2) \cdot (1 + 0,15) = 13800 \text{ руб.}$$

Страховой взнос после внедрения вычисляются по формуле 20.

$$O_{\text{ВН2}} = ЗП_2 \cdot 0,3, \quad (20)$$

$$O_{\text{ВН2}} = 13800 \cdot 0,3 = 4140 \text{ руб.}$$

Рассчитываем затраты, связанные с эксплуатацией ЭВМ после внедрения по формуле 21.

$$З_{\text{ЭВМ2}} = T_{2\text{час}} \cdot C_{\text{М-ч}}, \quad (21)$$

$$З_{\text{ЭВМ2}} = 70 \cdot 8 = 560 \text{ руб.}$$

Подставив соответствующие значения в формулу, получим:

$$C_2 = 13800 + 4140 + 560 + 1500 = 20000 \text{ руб.}$$

Таким образом, текущие затраты на содержание системы до внедрения разработанной программы для микроконтроллера составляют 47210 руб., после внедрения 20000 руб.

3.4 Расчет экономической целесообразности разработки и внедрения информационных технологий

Для разрабатываемого проекта расчет экономической эффективности производится исходя из следующих условий:

- годовые текущие затраты до внедрения автоматизированной системы, $C_1 = 47210$ руб.,
- годовые текущие затраты после внедрения системы, $C_2 = 20000$ руб.,
- горизонт расчета принимается исходя из срока использования разработки, $T = T_n = 3$ годам,
- шаг расчета равен одному году, $t = 1$ году,
- капитальные вложения равны затратам на создание системы, $K = 52684,8$ руб.,
- норма дисконта равна норме дохода на капитал, $E = 12\%$.

Ожидаемая условно-годовая экономия от внедрения системы рассчитывается по формуле 22.

$$\mathcal{E}_{\text{уг}} = C_1 - C_2 + \sum \mathcal{E}_i, \quad (22)$$

где $\mathcal{E}_{\text{уг}}$ – ожидаемая условно-годовая экономия, руб.,

C_1 – годовые текущие затраты до внедрения автоматизированной системы, руб.,

C_2 – годовые текущие затраты после внедрения системы, руб.,

$\sum \mathcal{E}_i$ – ожидаемый дополнительный эффект от различных факторов, руб.

Так как основным фактором, по которому производится расчет экономической эффективности от внедрения программы для микроконтроллера, является уменьшение получения данных о времени и дополнительный эффект не учитывается, то $\sum \mathcal{E}_i = 0$.

Подставив вычисленные выше значения в формулу, получим:

$$\mathcal{E}_{\text{уг}} = 47210 - 20000 = 27210 \text{ руб.}$$

где $\Delta_{уг}$ – ожидаемая условно–годовая экономия, руб.

Величина ожидаемого годового экономического эффекта от внедрения ИС рассчитывается по формуле 23.

$$\Delta_{г} = \Delta_{уг} - K \cdot E_{н}, \quad (23)$$

где $\Delta_{г}$ – ожидаемый годовой экономический эффект, руб.,

$\Delta_{уг}$ – ожидаемая условно-годовая экономия, руб.,

K – капитальные вложения (равны затратам на создание ИС), руб.,

$E_{н}$ – нормативный коэффициент экономической эффективности капитальных вложений.

Нормативный коэффициент экономической эффективности капитальных вложений определяется по формуле 24.

$$E_{н} = \frac{1}{T_{н}}, \quad (24)$$

где $T_{н}$ – нормативный срок окупаемости капитальных вложений, лет.

Подставив вычисленные выше значения в формулу, получим:

$$\Delta_{г} = 27210 - 52684,8 \cdot 0,33 = 9648,4 \text{ руб.}$$

Расчетный коэффициент экономической эффективности капитальных вложений рассчитывается по формуле 25.

$$E_{р} = \frac{\Delta_{уг}}{K}, \quad (25)$$

где $E_{р}$ – расчетный коэффициент экономической эффективности капитальных вложений,

$\Delta_{уг}$ – ожидаемая условно–годовая экономия, руб.,

K – капитальные вложения на создание системы, руб.

Подставив вычисленные выше значения в формулу, получим:

$$E_{р} = \frac{27210}{52684,8} = 0,5$$

Расчетный срок окупаемости капитальных вложений определяется по формуле 26.

$$T_p = \frac{1}{E_p}, \quad (26)$$

где E_p – коэффициент экономической эффективности капитальных вложений.

Подставив вычисленные выше значения в формулу, получим:

$$T_p = \frac{1}{0,5} = 2 \text{ год.}$$

Срок окупаемости без дисконтирования 2 года.

Чистый дисконтированный доход (ЧДД) определяется как сумма текущих эффектов за весь расчетный период, приведенная к начальному шагу, или как превышение интегральных результатов над интегральными затратами.

Если в течение расчетного периода не происходит инфляционного изменения цен или расчет производится в базовых ценах, то величина ЧДД для постоянной нормы дисконта вычисляется по формуле 27.

$$\text{ЧДД} = \sum_{t=1}^T (P_t - Z_t) \cdot \frac{1}{(1 + E)^t} - K, \quad (27)$$

где P_t – ожидаемые результаты от внедрения предложенной ИС, руб.,

Z_t – ожидаемые затраты (капитальные и текущие) на создание и эксплуатацию ИС, руб.,

$\text{Э}_t = (P_t - Z_t)$ – эффект, достигаемый на t -м шаге расчета,

K – капитальные вложения,

t – номер шага расчета ($t = 1, 2, 3$),

T – горизонт расчета,

E – постоянная норма дисконта, 12%.

$\text{Э}_t = (P_t - Z_t) = \text{Э}_{\text{уг}} = 27210$ руб. В том случае, если текущие затраты (Z_t) на весь срок использования разработки равны 0.

$t = 1, 2, 3$ год, т.к. предполагается, что результат от внедрения предложенной ИС будет с текущего года внедрения ИС.

Если ЧДД инвестиционного проекта положителен, то проект является эффективным (при данной норме дисконта).

Тогда суммарный чистый дисконтированный доход за весь горизонт расчета равен:

$$\text{ЧДД} = \text{Э}_1 \cdot \frac{1}{(1+E)} + \text{Э}_2 \cdot \frac{1}{(1+E)^2} + \text{Э}_3 \cdot \frac{1}{(1+E)^3} - K, \quad (28)$$

$$\text{ЧДД} = \frac{27210}{(1+0,12)} + \frac{27210}{(1+0,12)^2} + \frac{27210}{(1+0,12)^3} - 52684,8 = 12669 \text{ руб.}$$

Положительное значение чистого дисконтированного дохода, $\text{ЧДД} > 0$, свидетельствует о том, что инвестирование целесообразно и данная ИС может приносить прибыль в установленном объеме.

Индекс доходности (ИД) представляет собой отношение суммы приведенных эффектов к величине капитальных вложений и определяется по формуле 27.

$$\text{ИД} = \frac{1}{K} \sum_{t=1}^T (P_t - Z_t) \cdot \frac{1}{(1+E)^t}, \quad (29)$$

где K – величина капиталовложений или стоимость инвестиций.

$$\text{ИД} = \frac{65353,8}{52684,8} = 1,2$$

Инвестиции считаются эффективными, если индекс доходности выше единицы, $\text{ИД} > 1$, следовательно, инвестиции в данную ИС, эффективны.

Внутренняя норма доходности (ВНД):

при $E_1 \rightarrow \text{ЧДД}_1 > 0$

$E_2 \rightarrow \text{ЧДД}_2 < 0$

$$\text{ВНД} = E_1 + \frac{\text{ЧДД}_1}{\text{ЧДД}_1 - \text{ЧДД}_2} \cdot (E_2 - E_1), \quad (30)$$

при $E_1 \rightarrow \text{ЧДД}_1 > 0$

$E_2 \rightarrow \text{ЧДД}_2 > 0$

$$\text{ВНД} = E_1 + \frac{\text{ЧДД}_1}{\text{ЧДД}_1 + \text{ЧДД}_2} \cdot (E_2 - E_1) \quad (31)$$

$$E_1 = 0,11$$

$$ЧДД_1 = \frac{27210}{(1 + 0,11)} + \frac{27210}{(1 + 0,11)^2} + \frac{27210}{(1 + 0,11)^2} - 52684,8 = 13808 \text{ руб.}$$

$$E_1 = 0,13$$

$$ЧДД_2 = \frac{27210}{(1 + 0,13)} + \frac{27210}{(1 + 0,13)^2} + \frac{27210}{(1 + 0,13)^2} - 52684,8 = 11562 \text{ руб.}$$

$$E_1 \rightarrow ЧДД_1 > 0$$

$$E_2 \rightarrow ЧДД_2 > 0$$

$$ВНД = 0,11 + \frac{13808}{13808 + 11562} \cdot (0,13 - 0,11) = 0,12.$$

Таким образом, норма дисконта должна быть в пределах 11% – 13%.

Показатели экономической целесообразности разработки и внедрения программы для микроконтроллера сведены в таблицу 7.

Таблица 7 – Показатели экономической целесообразности разработки и внедрения программы для микроконтроллера

Наименование показателя	Значения
Затраты на разработку и внедрение ПП, руб.	52684,8
Ожидаемая экономия от внедрения ПП, руб.	27210
Чистый дисконтированный доход, руб.	12669
Индекс доходности	1,2
Внутренняя норма доходности	0,12
Дисконтированный срок окупаемости, лет	2
Срок морального старения, года	3

Произведенные расчеты свидетельствуют, что внедрение, разработанной в ВКР программы для микроконтроллера, позволит сократить временные затраты на получение данных о времени, что приведет к сокращению годовых текущих затрат на 27210 рублей

Опираясь на оценку экономической эффективности, можно сделать вывод о том, что разработка и внедрение предлагаемой программы для микроконтроллера является экономически обоснованной и целесообразной.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы были разработаны голографические часы. Все поставленные задачи были выполнены в полном объёме:

- выполнен анализ предметной области и изучены требования для проектируемого устройства,
- выполнен выбор элементной базы устройства, разработана принципиальная схема устройства, печатная плата, программный код и корпус устройства,
- рассмотрены основы охраны труда и промышленной экологии,
- проведён расчёт экономических параметров разработанного устройства.

Разработанное устройство может использоваться как частными пользователями, так и некоторыми организациями, для украшения помещений в качестве декора.

Данная разработка является основой, способной принимать множество трансформаций и модернизаций, различными путями, чтобы удовлетворить современным нуждам заказчика, при этом не обязательно использовать компоненты и программы, представленные в данной работе, так как существует множество вариаций и каждый может выбрать то, что необходимо конкретного для его задачи, при этом основная идея сохраняется.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Голография: история и применение [электронный ресурс] URL: <https://multiurok.ru/blog/golografiia-istoriia-i-primienieniie.html> - (дата обращения 5.03.2022г)
2. Технологии создания голограмм http://genius.pstu.ru/file.php/1/pupils_works_2017/Ahidova_Sofja.pdf
3. Будущее рядом, 3D голограммы [электронный ресурс] URL: <https://event-tech.org/blog/budushcheye-ryadom-3d-gologrammy/> - (дата обращения 5.03.2022г)
4. Что такое голограмма и где она используется [электронный ресурс] URL: <https://www.techcult.ru/technology/5215-chto-takoe-gologramma> - (дата обращения 5.03.2022г)
5. Дифракция света [электронный ресурс] URL: <https://light-fizika.ru/index.php/11-klass?layout=edit&id=152> - (дата обращения 5.03.2022г)
6. Дисперсия света. Интерференция [электронный ресурс] URL: <https://light-fizika.ru/index.php/11-klass?layout=edit&id=151> - (дата обращения 5.03.2022г)
7. Настольные электронные часы Uniel UTP-80 [электронный ресурс] URL: <https://goo.su/7nKg7> - (дата обращения 5.03.2022г)
8. Голографический проектор вентилятор 3D [электронный ресурс] URL: <https://goo.su/Qh1ClyH> - (дата обращения 5.03.2022г)
9. Gift Development / Светодиодные настольные [электронный ресурс] URL: <https://www.wildberries.ru/catalog/11817262/detail.aspx> - (дата обращения 5.03.2022г)
10. Зеркальные настольные часы с радио [электронный ресурс] URL: <https://goo.su/odOo8M> - (дата обращения 5.03.2022г)
11. Список микроконтроллеров ATmega. [электронный ресурс]. URL: <https://chipinfo.pro/mcu/avr/atmegalist.shtml> – (дата обращения 21.03.2022г).

12. Разработка электроники. О микроконтроллерах на пальцах [электронный ресурс]. URL: <https://habr.com/ru/post/445936/> – (дата обращения 18.02.2022г)
13. Микроконтроллеры. Устройство и особенности. [электронный ресурс]. URL: <https://electrosam.ru/glavnaja/slabotochnye-seti/oborudovanie/mikrokontrollery/> – (дата обращения 18.02.2022г)
14. Оценка условий труда на рабочем месте веб-дизайнера [электронный ресурс] URL: <http://www.refbzd.ru/viewreferat-1696-11.html> - (дата обращения 14.04.2022г)
15. Инструкция по охране труда для инженера по эксплуатации электротехнического оборудования [электронный ресурс] URL: <https://инструкция-по-охране-труда.рф/по-эксплуатации-электротехнического.html> - (дата обращения 13.04.2022г)
16. Вентилятор DEXP DX120T [электронный ресурс]. URL: <https://www.dns-shop.ru/product/e8834efc75e33330/ventilator-dexp-dx120t/characteristics/> – (дата обращения 21.03.2022г).
17. Вентилятор Aerocool Force 12 [электронный ресурс]. URL: <https://www.dns-shop.ru/product/c9d7a93960ec3332/ventilator-aerocool-force-12-acf3-fc0011011/> – (дата обращения 21.03.2022г).
18. Вентилятор 5Bites [электронный ресурс]. URL: <https://www.dns-shop.ru/product/544aecf1cc412ff4/ventilator-5bites-fb5010s-1213/characteristics/> – (дата обращения 21.03.2022г).
19. Анализ опасных и вредных факторов, воздействующих на программиста [электронный ресурс] URL: <https://pandia.ru/text/78/634/69969.php> – (дата обращения 13.04.2022г)
20. Артеменко С.В., Тихоненко Д.В. Утилизация компьютерной техники. 2020. №8. URL: <https://cyberleninka.ru/article/n/utilizatsiya-kompyuternoju-tehniki> (дата обращения: 25.03.2022)

Приложение А – Программный код

```
#include <EEPROM.h> //подключаем библиот. еепром и часов реального
времени
#include <iarduino_RTC.h>
iarduino_RTC time(RTC_DS3231);

#define up A0 //кнопка +
#define down A2 //кнопка -
#define center A1 //кнопка режим
#define keyOn 2 //Определение порта синхронизатора

byte timeON = 10; //Время работы - время одной ячейки отображения, ниже
ограничено 4-40мкс
byte adr = 0; //переменная адрес еепром
int ii = 0; //номер меню
int mn = 0; //промежуточные изменения единиц времени
int hr = 0;
int i = 1;
volatile int x = 1; //переменные для плавного смещения табло
volatile int tw = 0;
bool timeStart = true; //для смены табло
void setup() { //Ножки для светодиодов, с низу вверх.
  pinMode(9, OUTPUT); // 10 диод
  pinMode(8, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(3, OUTPUT);
```

```

pinMode(13, OUTPUT);
pinMode(12, OUTPUT);
pinMode(11, OUTPUT); // 1 диод
pinMode(keyOn, INPUT);
time.begin();
attachInterrupt(0, indicator, FALLING); //внешнее прерывание для индикации
timeON = EEPROM.read(0);
}
//Функция матрицы символа
void func(int ten, int nine, int eight, int seven, int six, int five, int four, int three, int
two, int one)
{
digitalWrite (11, one);
digitalWrite (12, two);
digitalWrite (13, three);
digitalWrite (3, four);
digitalWrite (4, five);
digitalWrite (5, six);
digitalWrite (6, seven);
digitalWrite (7, eight);
digitalWrite (8, nine);
digitalWrite (9, ten);
delayMicroseconds(timeON);
}
//матрица символов
void zero() {
func(0, 1, 1, 1, 1, 1, 1, 1, 1, 0);
func(1, 1, 0, 0, 0, 0, 0, 0, 1, 1);
func(1, 1, 0, 0, 0, 0, 0, 0, 1, 1);

```

```

func(0, 1, 1, 1, 1, 1, 1, 1, 1, 0);
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
void one() {
func(1, 1, 0, 0, 0, 0, 0, 0, 0, 0);
func(1, 1, 0, 0, 0, 0, 1, 1, 0, 0);
func(1, 1, 1, 1, 1, 1, 1, 1, 1, 1);
func(1, 1, 0, 0, 0, 0, 0, 0, 0, 0);
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
void two() {
func(1, 1, 1, 1, 1, 0, 0, 1, 1, 0);
func(1, 1, 0, 0, 1, 1, 0, 0, 1, 1);
func(1, 1, 0, 0, 1, 1, 0, 0, 1, 1);
func(1, 1, 0, 0, 0, 1, 1, 1, 1, 0);
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
void three() {
func(1, 1, 1, 0, 0, 0, 0, 1, 1, 1);
func(1, 1, 0, 0, 1, 1, 0, 0, 1, 1);
func(1, 1, 0, 0, 1, 1, 0, 0, 1, 1);
func(1, 1, 1, 1, 1, 1, 1, 1, 1, 1);
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
void four() {
func(0, 0, 0, 0, 0, 1, 1, 1, 1, 1);
func(0, 0, 0, 0, 1, 1, 0, 0, 0, 0);
func(0, 0, 0, 0, 1, 1, 0, 0, 0, 0);
}

```

```

func(1, 1, 1, 1, 1, 1, 1, 1, 1, 1);
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
void five() {
func(0, 1, 1, 0, 1, 1, 1, 1, 1, 1);
func(1, 1, 0, 0, 1, 1, 0, 0, 1, 1);
func(1, 1, 0, 0, 1, 1, 0, 0, 1, 1);
func(0, 1, 1, 1, 1, 0, 0, 0, 1, 1);
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
void six() {
func(0, 1, 1, 1, 1, 1, 1, 1, 1, 0);
func(1, 1, 0, 0, 1, 1, 0, 0, 1, 1);
func(1, 1, 0, 0, 1, 1, 0, 0, 1, 1);
func(0, 1, 1, 1, 1, 0, 0, 1, 1, 0);
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
void seven() {
func(0, 0, 0, 0, 0, 0, 0, 1, 1, 1);
func(0, 0, 0, 0, 0, 0, 0, 0, 1, 1);
func(0, 0, 0, 0, 0, 0, 0, 0, 1, 1);
func(1, 1, 1, 1, 1, 1, 1, 1, 1, 1);
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
void eight() {
func(0, 1, 1, 1, 0, 0, 1, 1, 1, 0);
func(1, 1, 0, 0, 1, 1, 0, 0, 1, 1);
func(1, 1, 0, 0, 1, 1, 0, 0, 1, 1);
}

```

```

func(0, 1, 1, 1, 0, 0, 1, 1, 1, 0);
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
void nine() {
func(0, 1, 1, 0, 0, 1, 1, 1, 1, 0);
func(1, 1, 0, 0, 1, 1, 0, 0, 1, 1);
func(1, 1, 0, 0, 1, 1, 0, 0, 1, 1);
func(0, 1, 1, 1, 1, 1, 1, 1, 1, 0);
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
void point() {
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
func(0, 0, 1, 1, 0, 0, 1, 1, 0, 0);
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
void ppoint() {
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
func(1, 1, 0, 0, 0, 0, 0, 0, 0, 0);
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
//матрицы букв
void pp() {
func(1, 1, 1, 1, 1, 1, 1, 1, 1, 1);
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 1);
func(1, 1, 1, 1, 1, 1, 1, 1, 1, 1);
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}

```

```

}
void nn() {
    func(1, 1, 1, 1, 1, 1, 1, 1, 1, 1);
    func(0, 0, 0, 0, 1, 0, 0, 0, 0, 0);
    func(1, 1, 1, 1, 1, 1, 1, 1, 1, 1);
    func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
void vv() {
    func(1, 1, 1, 1, 1, 1, 1, 1, 1, 1);
    func(1, 0, 0, 0, 0, 1, 0, 0, 0, 1);
    func(1, 0, 0, 0, 0, 1, 1, 1, 1, 1);
    func(1, 1, 1, 1, 1, 1, 0, 0, 0, 0);
    func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
void tt() {
    func(0, 0, 0, 0, 0, 0, 0, 0, 0, 1);
    func(1, 1, 1, 1, 1, 1, 1, 1, 1, 1);
    func(0, 0, 0, 0, 0, 0, 0, 0, 0, 1);
    func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
void ss() {
    func(1, 1, 1, 1, 1, 1, 1, 1, 1, 1);
    func(1, 0, 0, 0, 0, 0, 0, 0, 0, 1);
    func(1, 0, 0, 0, 0, 0, 0, 0, 0, 1);
    func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
void rr() {
    func(1, 1, 1, 1, 1, 1, 1, 1, 1, 1);

```

```

func(0, 0, 0, 0, 0, 1, 0, 0, 0, 1);
func(0, 0, 0, 0, 0, 1, 1, 1, 1, 1);
func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
void ch() {
    func(0, 0, 0, 0, 0, 1, 1, 1, 1, 1);
    func(0, 0, 0, 0, 0, 1, 0, 0, 0, 0);
    func(1, 1, 1, 1, 1, 1, 1, 1, 1, 1);
    func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
void bb() {
    func(1, 1, 1, 1, 1, 1, 1, 1, 1, 1);
    func(1, 0, 0, 0, 0, 1, 0, 0, 0, 1);
    func(1, 1, 1, 1, 1, 1, 0, 0, 0, 1);
    func(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
//функции расчета и отображения
void secDec () { //все отображение бьется на декады и единицы
    switch (time.seconds / 10) {
        case 0: zero(); break;
        case 1: one(); break;
        case 2: two(); break;
        case 3: three(); break;
        case 4: four(); break;
        case 5: five(); break;
    }
}
void secUni () {

```

```

switch (time.seconds % 10) {
    case 0: zero(); break;
    case 1: one(); break;
    case 2: two(); break;
    case 3: three(); break;
    case 4: four(); break;
    case 5: five(); break;
    case 6: six(); break;
    case 7: seven(); break;
    case 8: eight(); break;
    case 9: nine(); break;
}
}

void minDec () {
    switch (time.minutes / 10) {
        case 0: zero(); break;
        case 1: one(); break;
        case 2: two(); break;
        case 3: three(); break;
        case 4: four(); break;
        case 5: five(); break;
    }
}

void minUni () {

    switch (time.minutes % 10) {
        case 0: zero(); break;
        case 1: one(); break;

```

```

    case 2: two(); break;
    case 3: three(); break;
    case 4: four(); break;
    case 5: five(); break;
    case 6: six(); break;
    case 7: seven(); break;
    case 8: eight(); break;
    case 9: nine(); break;
}
}
void hourDec () {
    switch (time.Hours / 10) {
        case 0: zero(); break;
        case 1: one(); break;
        case 2: two(); break;
    }
}
void hourUni () {
    switch (time.Hours % 10) {
        case 0: zero(); break;
        case 1: one(); break;
        case 2: two(); break;
        case 3: three(); break;
        case 4: four(); break;
        case 5: five(); break;
        case 6: six(); break;
        case 7: seven(); break;
        case 8: eight(); break;

```

```

        case 9: nine(); break;
    }
}
void ddDec () {
    switch (time.day / 10) {
        case 0: zero(); break;
        case 1: one(); break;
        case 2: two(); break;
        case 3: three(); break;
    }
}
void ddUni () {
    switch (time.day % 10) {
        case 0: zero(); break;
        case 1: one(); break;
        case 2: two(); break;
        case 3: three(); break;
        case 4: four(); break;
        case 5: five(); break;
        case 6: six(); break;
        case 7: seven(); break;
        case 8: eight(); break;
        case 9: nine(); break;
    }
}
void meUni () {
    switch (time.month) {
        case 1: zero(); one(); break;

```

```

    case 2: zero(); two(); break;
    case 3: zero(); three(); break;
    case 4: zero(); four(); break;
    case 5: zero(); five(); break;
    case 6: zero(); six(); break;
    case 7: zero(); seven(); break;
    case 8: zero(); eight(); break;
    case 9: zero(); nine(); break;
    case 10: one(); zero(); break;
    case 11: one(); one(); break;
    case 12: one(); two(); break;
}
}
void yyDec () {
    switch (time.year / 10) {
        case 0: zero(); break;
        case 1: one(); break;
        case 2: two(); break;
        case 3: three(); break;
        case 4: four(); break;
        case 5: five(); break;
        case 6: six(); break;
    }
}
void yyUni () {
    switch (time.year % 10) {
        case 0: zero(); break;
        case 1: one(); break;

```

```

    case 2: two(); break;
    case 3: three(); break;
    case 4: four(); break;
    case 5: five(); break;
    case 6: six(); break;
    case 7: seven(); break;
    case 8: eight(); break;
    case 9: nine(); break;
}
}
void wUni () {
    switch (time.weekday % 10) {
        case 0: vv(); ss(); break;
        case 1: pp(); nn(); break;
        case 2: vv(); tt(); break;
        case 3: ss(); rr(); break;
        case 4: ch(); tt(); break;
        case 5: pp(); tt(); break;
        case 6: ss(); bb(); break;
    }
}

```

//общие функции вывода времени\даты по порядку отображения

```

void ind() {
    hourDec ();
    hourUni ();
    point ();
    minDec ();
    minUni ();
}

```

```

point ();
secDec ();
secUni ();
}
void indd() {
    wUni ();
    ppoint();
    ddDec ();
    ddUni ();
    ppoint();
    meUni ();
    ppoint();
    yyDec ();
    yyUni ();
}
//функции изменений времени
void mig() { //Мигаем светодиодом в подТВ. изменения единицы
    digitalWrite (10, 1);
    delay(300);
    digitalWrite (10, 0);
    delay(300);
}
void minCh() { //МИН+-
    while (ii == 1) {
        digitalWrite (9, 1);
        if (digitalRead(up) == 1) {
            time.getTime();//получаем текущее время
            mn = time.minutes;

```

```

    mn = mn + 1; //изменяем его на единицу
    if (mn > 59) { //тут ограничение для минут, в другой ф-ях ограничение
другое
        mn = 0;
    }
    mig();
    time.setTime(-1, mn); //уст. время
}
if (digitalRead(down) == 1) {
    time.getTime();
    mn = time.minutes;
    mn = mn - 1;
    if (mn < 0) {
        mn = 59;
    }
    mig();
    time.setTime(-1, mn);
}
if (digitalRead(center) == 1) { //переходим на "меню" вверх
    ii = ii + 1;
    delay(300);
}
}
}
void hourCh() { //час+-
    while (ii == 2) {
        digitalWrite(9, 0);
        digitalWrite(8, 1);
        if (digitalRead(up) == 1) {

```

```

time.getTime();
hr = time.Hours;
hr = hr + 1;
if (hr > 23) {
    hr = 0;
}
mig();
time.setTime(-1, -1, hr);
}
if (digitalRead(down) == 1) {
    time.getTime();
    hr = time.Hours;
    hr = hr - 1;
    if (hr < 0) {
        hr = 23;
    }
    mig();
    time.setTime(-1, -1, hr);
}
if (digitalRead(center) == 1) {
    ii = ii + 1;
    delay(300);
}
}
}
void dayCh() { //день+-
    while (ii == 4) {
        digitalWrite(7, 0);
    }
}

```

```

digitalWrite (6, 1);
if (digitalRead(up) == 1) {
    time.getTime();
    mn = time.day;
    mn = mn + 1;
    if (mn > 31) {
        mn = 1;
    }
    mig();
    time.setTime(-1, -1, -1, mn);
}
if (digitalRead(down) == 1) {
    time.getTime();
    mn = time.day;
    mn = mn - 1;
    if (mn < 1) {
        mn = 31;
    }
    mig();
    time.setTime(-1, -1, -1, mn);
}
if (digitalRead(center) == 1) {
    ii = ii + 1;
    delay(300);
}
}
}
void mesCh() { //месяц+-

```

```

while (ii == 5) {
    digitalWrite (6, 0);
    digitalWrite (5, 1);
    if (digitalRead(up) == 1) {
        time.getTime();
        mn = time.month;
        mn = mn + 1;
        if (mn > 12) {
            mn = 1;
        }
        mig();
        time.setTime(-1, -1, -1, -1, mn);
    }
    if (digitalRead(down) == 1) {
        time.getTime();
        mn = time.month;
        mn = mn - 1;
        if (mn < 1) {
            mn = 12;
        }
        mig();
        time.setTime(-1, -1, -1, -1, mn);
    }
    if (digitalRead(center) == 1) {
        ii = ii + 1;
        delay(300);
    }
}

```

```

}
void yearCh() { //год+-
  while (ii == 6) {
    digitalWrite(5, 0);
    digitalWrite(4, 1);
    if (digitalRead(up) == 1) {
      time.getTime();
      mn = time.year;
      mn = mn + 1;
      if (mn > 98) {
        mn = 1;
      }
      mig();
      time.setTime(-1, -1, -1, -1, -1, mn);
    }
    if (digitalRead(down) == 1) {
      time.getTime();
      mn = time.year;
      mn = mn - 1;
      if (mn < 1) {
        mn = 99;
      }
      mig();
      time.setTime(-1, -1, -1, -1, -1, mn);
    }
    if (digitalRead(center) == 1) {
      ii = ii + 1;
      delay(300);
    }
  }
}

```

```

    }
}
}
void weekdayCh() { //день недели+-
while (ii == 7) {
    digitalWrite(4, 0);
    digitalWrite(3, 1);
    if (digitalRead(up) == 1) {
        time.getTime();
        mn = time.weekday;
        mn = mn + 1;
        if (mn > 6) {
            mn = 1;
        }
        mig();
        time.setTime(-1, -1, -1, -1, -1, -1, mn);
    }
    if (digitalRead(down) == 1) {
        time.getTime();
        mn = time.weekday;
        mn = mn - 1;
        if (mn < 0) {
            mn = 6;
        }
        mig();
        time.setTime(-1, -1, -1, -1, -1, -1, mn);
    }
    if (digitalRead(center) == 1) {

```

```

ii = ii + 1;
delay(300);
if (ii > 7) {
    ii = 0;
    digitalWrite(3, 0);
}
}
}

void timeonCh() { //ширина символов +- 4 мкс
while (ii == 3) {
    digitalWrite(8, 0);
    digitalWrite(7, 1);
    if (digitalRead(up) == 1) {
        timeON = timeON + 4; //4 -на сколько мкс изменится ширина ячейки
        СИМВОЛА
        if (timeON >= 40) { //40 - ограничение на макс. ширину ячейки.
            timeON = 4; //при превышении 40мкс устанавливаем минимальное -
            4 мкс. Самое минимальное 1 мкс.
        }
        EEPROM.write(adr, timeON); //записываем в память значение ширины
        ячейки
        mig();
    }
    if (digitalRead(down) == 1) {
        timeON = timeON - 4;
        if (timeON <= 0) {
            timeON = 40;
        }
    }
}
}
}

```

```

EEPROM.write(adr, timeON);
mig();
}
if (digitalRead(center) == 1) {
    timeON = EEPROM.read(adr);
    ii = ii + 1;
    delay(300);
}
}
}
//=====
void loop() {
    time.getTime(); //считаем время
    if (digitalRead(center) == 1) { //Меню выбора часов-минут
        ii = ii + 1;
        delay(300);
        if (ii > 7) {
            ii = 0;
            digitalWrite(3, 0);
        }
    }
    minCh();
    hourCh();
    timeonCh();
    dayCh();
    mesCh();
    yearCh();
    weekdayCh();
}

```

```

if (time.seconds / 10 == 1) { //переключаем табло, на 10сек раз в 1 минуту
    timeStart = false;
}
else {
    timeStart = true;
}
}
void indicator() { //отображение (ф-я прерывания)
    interrupts();
    delayMicroseconds(tw); //задержка плавного перемещения табло
    if (timeStart == true) {
        ind();
    }
    else {
        indd();
    }
    if (tw == 8000) {
        x = -40; //8000 - диапазон смещения табло, -40 - скорость смещения
    }
    if (tw == 0) {
        x = 40;
    }
    tw = tw + x;
}

```