|  |  |
| --- | --- |
|  | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ   Радиоэлектроника и лазерная техника (РЛ)

КАФЕДРА   Радиоэлектронный системы и устройства (РЛ1)

**ОТЧЕТ ПО технологической ПРАКТИКЕ**

Студент   Чекановский Сергей Александрович

(Фамилия, имя, отчество)

Группа   РЛ1-61

Тема практики  Передача отсчетов цифрового радиосигнала с ПЛИС на ПК и их графическое отображение. Управление формирователем радиосигналов на ПЛИС с помощью микроконтроллера

Название предприятия  АО «Российские космические системы»

Руководитель практики

от МГТУ им. Н.Э. Баумана \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Подпись, дата) (И.О.Фамилия)

Руководитель практики

от  **АО «Российские**

**космические системы»**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Подпись, дата) (И.О.Фамилия)

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**      Чекановский С.А.

(Подпись, дата) (И.О.Фамилия)

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*2021г.*

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

Введение

Цель практики - передача отсчетов цифрового радиосигнала с ПЛИС на ПК и их графическое отображение. Управление формирователем радиосигналов на ПЛИС с помощью микроконтроллера.

Основное назначение микроконтроллеров – это управление различными электронными устройствами. Таким образом, они применяются не только в персональных компьютерах, но и почти во всей бытовой технике, автомобилях, телевизорах, промышленных роботах, даже в военных радиолокаторах.

Можно сказать, что микроконтроллер это универсальный инструмент управления электронными устройствами, причем алгоритм управления вы закладываете в него сами и можете в любое время его поменять в зависимости от задачи, возложенной на микроконтроллер.

В настоящее время существует множество различных платформ и семейств микроконтроллеров, однако назначение, применение и суть их функционирования почти одинакова.

В рамках производственной практики будет использоваться микроконтроллер ATmega328, который входит в состав платы Arduino. Выбор сделан в пользу данной платы ввиду легкости написания программ и популярности используемого языка.

**Раздел 1. Теоретическая часть**

* 1. **Общая информация об используемом микроконтроллере**

Плата Arduino Uno  – самое популярное и самое доступное устройство Arduino. В ее основе лежит чип ATmega – в последней ревизии Arduino Uno R3 – это ATmega328 (хотя на рынке можно еще встретить варианты платы UNO с ATmega168).

Характеристики платы представлены на рис. 1.

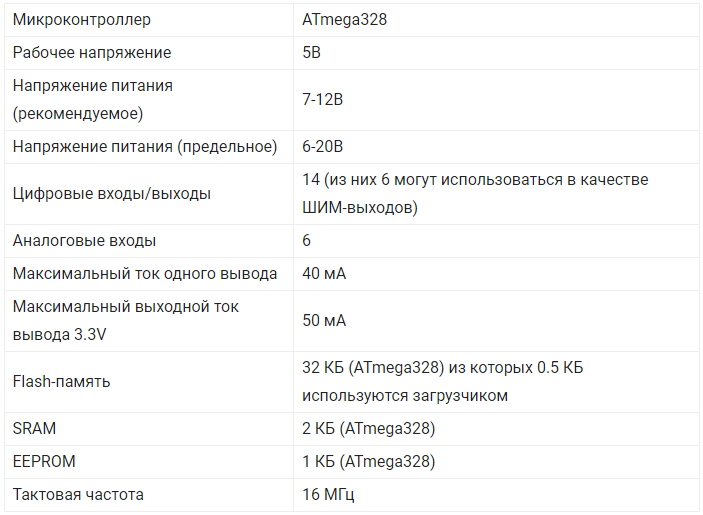


Рис. 1.

Плата выглядит следующим образом (рис. 2):

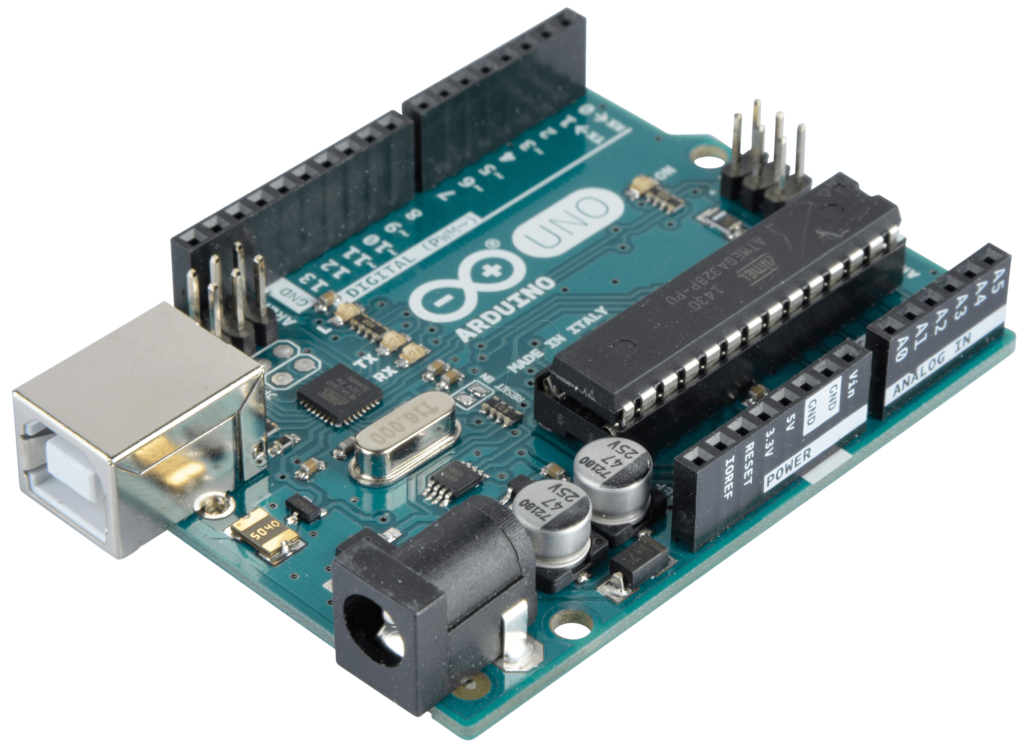


Рис. 2.

Распиновка платы (рис. 3):

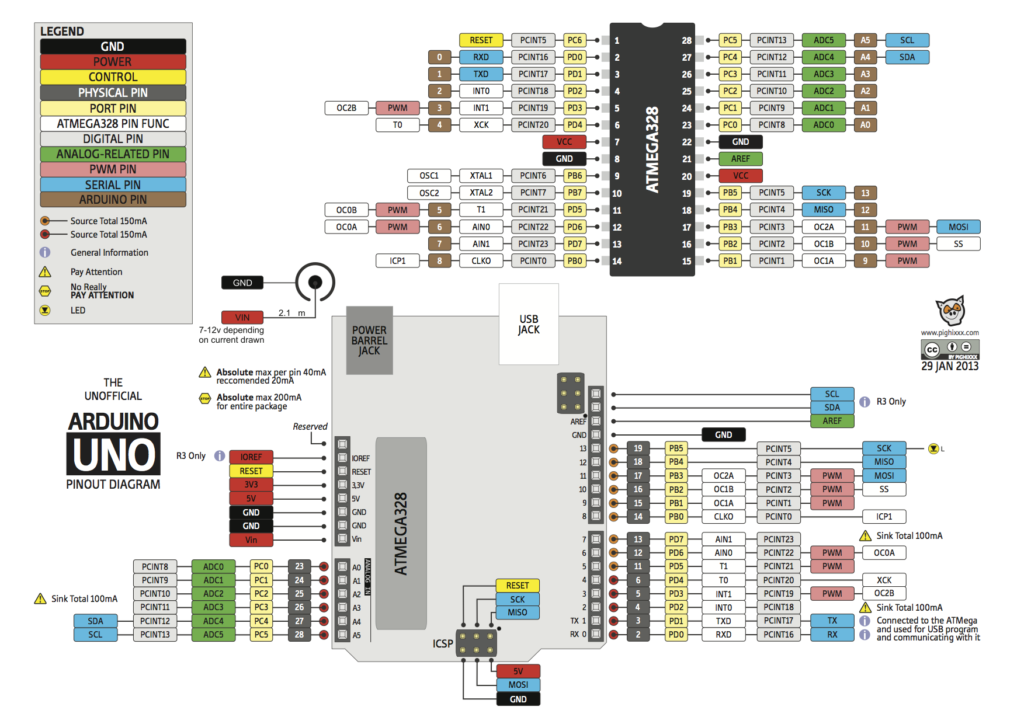


Рис. 3.

Принципиальная схема (рис. 4):

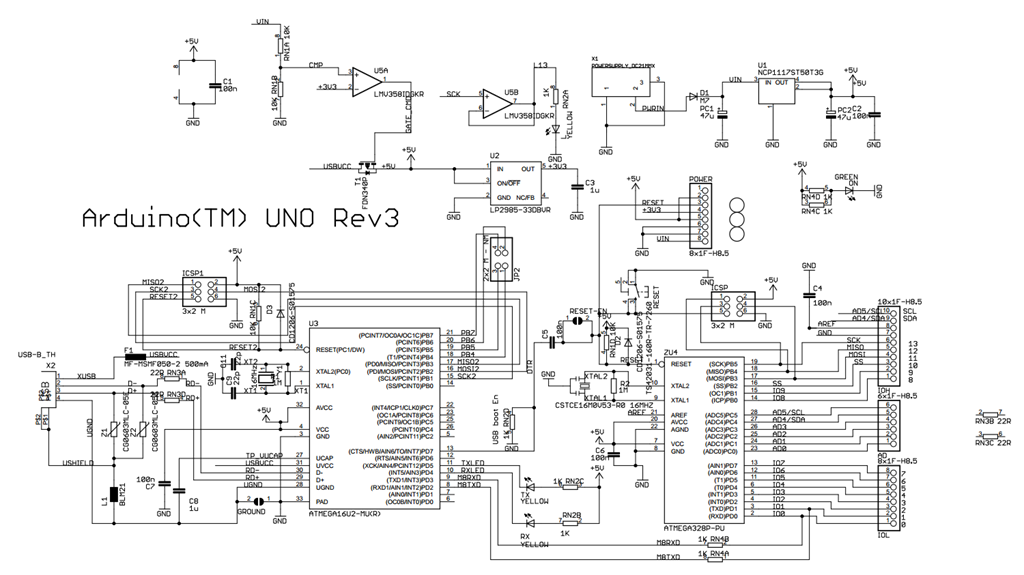


Рис. 4.

Распиновка микроконтроллера (рис. 5):

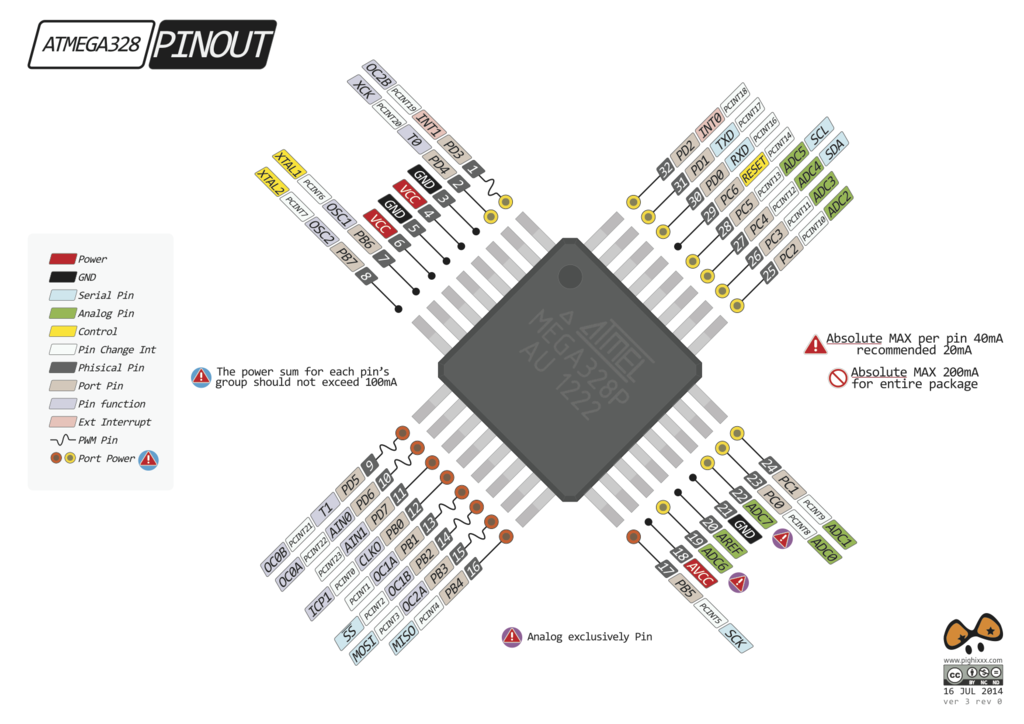


Рис. 5.

Пины Ардуино используются для подключения внешних устройств и могут работать как в режиме входа (INPUT), так и в режиме выхода (OUTPUT).  К каждому входу может быть подключен встроенный резистор 20-50 кОм с помощью выполнения команды pinMode () в режиме INPUT\_PULLUP. Допустимый ток на каждом из выходов – 20 мА, не более 40 мА в пике.  
Для удобства работы некоторые пины совмещают в себе несколько функций:

* Пины 0 и 1  – контакты UART (RХ и TX соответственно) .
* Пины c 10 по 13 – контакты SPI (SS, MOSI, MISO и SCK соответственно)
* Пины A4 и A5 – контакты I2C (SDA и SCL соответственно).

Подключение любых устройств к плате осуществляется путем присоединения к контактам, расположенным на плате контроллера: одному из цифровых или аналоговых пинов или пинам питания.

Рабочее напряжение платы Ардуино Уно – 5 В. На плате установлен стабилизатор напряжения, поэтому на вход можно подавать питание с разных источников. Кроме этого, плату можно запитывать с USB – устройств. Источник питания выбирается автоматически.

* Питание от внешнего адаптера, рекомендуемое напряжение от 7 до 12 В. Максимальное напряжение 20 В, но значение выше 12 В с высокой долей вероятности быстро выведет плату из строя. Напряжение менее 7 В может привести к нестабильной работе, т.к. на входном каскаде может запросто теряться 1-2 В. Для подключения питания может использоваться встроенный разъем DC 2.1 мм или напрямую вход VIN для подключения источника с помощью проводов.
* Питание от USB-порта компьютера.
* Подача 5 В напрямую на пин 5V. В этом случае обходится стороной входной стабилизатор  и даже малейшее превышение напряжения может привести к поломке устройства.

Плата Uno по умолчанию поддерживает три типа памяти:

* Flash – память объемом 32 кБ. Это основное хранилище для команд. Когда вы прошиваете контроллер своим скетчем, он записывается именно сюда. 2кБ из данного пула памяти отводится на bootloader- программу, которая занимается инициализацией системы, загрузки через USB и запуска скетча.
* Оперативная SRAM память объемом  2 кБ. Здесь по умолчанию хранятся переменные и объекты, создаваемые в ходе работы программы. Память энергозависимая, при выключении питания все данные сотрутся.
* Энергонезависимая память (EEPROM) объемом 1кБ. Здесь можно хранить данные, которые не сотрутся при выключении контроллера. Но процедура записи и считывания EEPROM требует использования дополнительной библиотеки, которая доступна в Arduino IDE по-умолчанию. Также нужно помнить об ограничении циклов перезаписи, присущих технологии EEPROM. Этот тип памяти не будет использоваться в рамках практики.

Для программирования платы можно установить Arduino IDE с официального сайта Arduino, но для данной практики был выбран более продвинутой вариант работы с платой через плагин для Visual Studio Code под названием PlatformIO. Visual Studio Code предоставляет более широкие возможности для работы с кодом и является гораздо более удобным инструментом чем оригинальная IDE.

* 1. **Процесс прошивки платы**

В микроконтроллер записывается программа, называемая прошивкой, которая позволяет получать сигналы с датчиков, обрабатывать нажатия кнопок, общаться с различными устройствами через интерфейсы, управлять исполнительными процессами.

Обычно прошивка записывается в кристалл микроконтроллера при помощи специальных устройств, называемых программаторами. Для разных микроконтроллеров существуют различные программаторы – от специализированных до универсальных. Важным отличием Ардуино от других контроллеров является возможность залить прошивку через обычный USB кабель. Это достигается при помощи специальной программы – загрузчика (Bootloader).

Загрузить прошивку в МК семейства Mega (а именно такая стоит в ардуине) можно разными способами:

-Последовательным программатором по интерфейсу SPI

-Через интерфейс JTAG

-Параллельным («высоковольтным») программатором.

Во всех этих случаях необходим программатор, а в ардуино всё прошивается обычным USB. Разработчики использовали одну удобную функцю МК Atmega под названием «самопрограммирование». Если в определённую область памяти программ Atmega прошить специальную программу-загрузчик (далее бутлоадер) то можно заставить её прошивать саму себя по любому имеющемуся на борту интерфейсу – I2C, UART или SPI. В ардуино используется именно UART — в USB версиях плат стоит преобразователь USB-to-COM (обычно ft232), а на COM— преобразователь уровней COM-to-TTL (max232 или на дискретных транзисторах)

Работает это следующим образом – когда Arduino перезагружается — запускается бутлоадер (если он есть). Он посылает импульсы на цифровой вывод 13 (к данному выводу подключён светодиод — он будет мигать) — это означает, что бутлоадер установлен и работает.

Далее бутлоадер ждёт команды или данные от компьютера. Обычно, это скетч, который бутлоадер записывает в flash-память МК. Затем, бутлоадер запускает свежезагруженную программу на выполнение.

Если никаких данных не поступает — бутлоадер запускает предыдущую программу, которая была загружена в микроконтроллер.

Если же никакой программы нет, то бутлоадер запускает себя снова (цикл повторяется).

* 1. **Возможные сбои в работе микроконтроллера**

В рамках практики не предусмотрена работа контроллера в экстремальных

условиях, однако по заданию следует рассмотреть различные факторы, которые могут помешать работе устройства.

Механизмы повреждения интегральных микросхем:

Анализ экспериментальных исследований показывает, что из-за сложности регистрации быстро протекающих процессов и изменений в структуре полупроводниковых элементов, происходящих при воздействии радиоимпульса, практически все авторы ограничиваются описанием условий возникновения и констатацией внешних особенностей отказа, что, конечно, не позволяет точно установить физическую причину повреждений. Весьма вероятно, что на высоких частотах действуют те же механизмы, что и в режиме видеоимпульса, однако эти механизмы (так же детально и подробно) не идентифицированы.

Наиболее распространенными механизмами повреждения ИМС считаются вторичный пробой, плавление металлизации, пробой диэлектрика. Рассмотрим их некоторые особенности.

Вторичный (тепловой) пробой в режиме коротких высокочастотных импульсов связан с попаданием рабочей точки в область электрического пробоя (на обратной ветви ВАХ), резким увеличением тока через прибор, значительным тепловыделением в области p-n перехода и последующим тепловым повреждением структуры. Чаще всего наблюдается проплавление локальных областей эмиттера входных транзисторов и p-n переходов защитных диодов.

В микросхемах, изготовленных по КМОП технологии, еще одной причиной повреждения может быть пробой окисной пленки под затвором. Характерное поле пробоя пленки SiO2 составляет 5×106 В/см, что при типичной толщине пленки 500…1000 Å дает напряжение пробоя 25…50 В.

Электрический пробой диэлектрика имеет место лишь в совершенных пленках, не имеющих дефектов. При наличии пор в окисле, через которые возможно шунтирование затвора электрода с подложкой, наблюдается тепловой пробой. Тепловой пробой происходит при напряжениях, на 5…10 В меньших напряжения пробоя диэлектрика. Часто первый электрический пробой не вызывает катастрофического отказа окисла пленки в ИМС, однако он оставляет структурно слабые точки на пленке. При повторном пробое эти точки продолжают ослаблять пленку, работающую до тех пор, пока пленка не пробивается.

Все механизмы имеют низкие пороги повреждений из-за малых геометрических размеров характерных областей в ИМС и наличия дефектов.

Характер процессов и механизмов повреждения интегральных микросхем существенным образом зависит как от параметров микросхем (размеров активных элементов, чипа, корпуса ИМС, длины ее выводов, их взаимного расположения, ориентации относительно вектора напряженности электрического поля), так и от амплитудно-временных характеристик излучения (интенсивности излучения, несущей частоты, длительности и частоты повторения импульсов, времени облучения). Это является одной из причин существенного разброса в результатах, приводимых разными авторами.

Повреждение микросхем при облучении СВЧ излучением может происходить в результате комбинации повреждений p-n переходов и металлизации. Как правило, из строя выходят входные и выходные элементы.

Так, в результате воздействия на микросхемы короткими радиоимпульсами (длительностью t = 20…30 нс и длиной волны l = 10 см) наблюдалось полное разрушение области p-n-перехода, расплавление и разрыв металлизации для ТТЛ микросхем при напряженности электрического поля Е = 1,4…1,8 кВ/см и для МОП микросхем при Е = 0,1…4 кВ/см

В результате анализа образцов ИМС, подвергнутых СВЧ воздействию, было найдено, что 90% биполярных и 63% КМОП ИМС выходят из строя из-за повреждения металлизации. Плавление чаще всего наблюдалось в области контактных площадок и поворотов токоведущих линий. В области плавления, как правило, имеется повышенная концентрация источников тепловыделения. Повреждения металлизации проявлялись в местах наибольшего механического напряжения в структуре: на ступеньках окисла и вокруг контактных площадок.

В ТТЛ, ТТЛШ микросхемах (на основе многоэмиттерного биполярного транзистора) наблюдалось повреждение защитных диодов, т.к. они имели минимальные размеры. Другие повреждения биполярных ИМС были обусловлены дефектами металлизации и межслойной изоляции. Повреждение ИМС, выполненных по современной технологии, вызывались разрушением тонкопленочных резисторов в коллекторных цепях выходных транзисторов и пробоем переходов эмиттер-база фазоинверсных каскадов.

При Em>10 кв/м начинаются деградационные явления в микроструктурных элементах микросхем, каковыми являются в основном локальные прожоги контактных площадок.

Анализ кристаллов, вышедших из строя вследствие импульсного воздействия, позволил выявить виды отказов современных микроконтроллеров. Основной причиной их отказов является прожог металлизации - 60%. В свою очередь из этих 60% отказов половина произошла вследствие прожога контактных площадок, а другая половина - вследствие прожога токопроводящих дорожек.

Было замечено, что 30% катастрофических отказов микросхем наступает вследствие одновременного теплового разрушения проводящих дорожек и полупроводниковых приборов, а 10% микросхем выходят из строя вследствие выгорания активных полупроводниковых микроструктурных элементов ИМС.

Для радиоизлучения (как непрерывного, так и импульсного различной длительности) проведено достаточно много исследований по воздействию на материалы и приборные структуры. И было установлено, что на стойкость интегральных микросхем существенно влияют параметры радиоизлучения (несущая частота, длительность и частота повторения импульсов, ориентация относительно электромагнитных полей) и конструктивно-технологические особенности ИМС (технология изготовления, топология ИМС, тип корпуса, выводов и т.п.).

Описанные результаты указывают на существенную роль преобразований точечно-дефектной структуры кристаллов, токоведущих линий, которые могут стимулироваться при воздействии радиоимпульсов. А развитие электрических пробоев (лавинного, туннельного, а также пробоя подзатворного диэлектрика) и тепловых процессов, ускоряют деградацию параметров и приводят к необратимым отказам в полупроводниковых приборах (повреждение p-n переходов, подзатворного диэлектрика и токоведущих линий).

Установлено, что физические механизмы деградационных изменений параметров и повреждения микросхем “запускаются” только при превышении определенного уровня мощности радиоимпульса.

Деградационные эффекты в импульсных полях проявляются при существенно более низких уровнях энергии, чем в стационарных полях.

Максимальные воздействия проявляются в импульсных полях, создающих наибольшие градиенты электрических полей и температуры.

Наименее устойчивыми к облучению являются неоднородные полупроводниковые структуры, содержащие встроенные электрические поля и границы раздела диэлектрика и металла.

Все результаты исследований стойкость интегральных микросхем в полях радиоизлучения носят статистический характер. Даже на идентичных микросхемах деградация параметров и выход из строя регистрировался при разных параметрах радиоизлучения (отличались интенсивность или время выхода из строя). Это связано, в частности, с различием в таких слабо контролируемых параметрах ИМС, как число и распределение дефектов внутри p-n переходов и других элементах микросхем.

При теоретической оценке уязвимости ИМС к воздействию радиоизлучения в широком диапазоне параметров используют математическое моделирование с использованием современных вычислительных средств. Однако точное положение районов повреждения в реальной конструкции микросхем немного отличается от расчетных, это часто заметено на исследованных устройствах и основано на допусках при производстве ИМС.

* 1. **Особенности монтажа микросхем на печантую плату**

Печатный монтаж состоит в закреплении элементов относительно плоскости платы в определенной пространственной ориентации и упорядоченном плоскостном расположении всех соединительных проводников и контактных площадок для их пайки. Роль проводников печатной платы выполняют участки тонкой медной фольги нанесенной на изоляционное основание платы.

Печатный монтаж РЭА по сравнению с объемным монтажом имеет следующие преимущества: значительное снижение трудоемкости за счет механизации и автоматизации сборочно-монтажных работ; повторяемость параметров от образца к образцу за счет идентичности форм и размеров печатных проводников; сокращение объема контрольно-испытательных операций и их автоматизация; повышение надежности изделий из-за сокращения числа паек и уменьшения ошибок при монтаже за счет автоматизации монтажно-сборочных и контрольных операций. Помимо этого печатный монтаж имеет другие достоинства: печатные проводники выдерживают в пять раз большую плотность тока, чем объемные; упрощается процесс поиска неисправностей; уменьшается масса изделия за счет ликвидации деталей промежуточного крепления элементов и проводов.

Наибольшее применение для изготовления печатных плат получили фальгированные диэлектрические материалы (гетинакс, стеклотекстолит), представляющие собой диэлектрик с нанесенной на него с одной или двух сторон медной фольгой. Для изготовления микромодулей, микросхем и микросборок применяют печатные платы (подложки) из эластичных полимеров, полиэфирных пленок, керамики и стекла. Формовку выводов и установку элементов на печатные платы следует производить в соответствии с требованиями: расстояние от корпуса элемента ГОСТа иди ТУ на элемент. При отсутствии этих сведений стандартом приняты следующие расстояния: от корпуса до места пайки не менее 2,5 мм; от корпуса до оси изогнутого вывода на менее 2 мм. Формовку круглых или планарных выводов необходимо производить при помощи технологической оснастки, исключающей механические нагрузки на места крепления ввода и вывода. Корпуса элементов должны располагаться параллельно или перпендикулярно друг другу. Предпочтительное расположение элементов - рядовое. Навесные элементы крепятся к печатной плате с помощью собственных выводов. В случае необходимости применяют дополнительное механическое крепление. Установку элемента с зазором между его корпусом и платой используют при двустороннем монтаже; при этом печатные проводники могут располагаться под навесным элементом. Лучшим способом с точки зрения восприятия механических нагрузок является установка элементов вплотную к плате, выполняемая с помощью собственных выводов и дополнительного крепления за корпус при помощи проволочных скоб, которые впаиваются в отверстия платы.

Маркировка электрорадиоэлементов должна быть нанесена в соответствии с их обозначениями в электрических принципиальных схемах. Разрешается производить маркировку на самих элементах, если это не повлияет на их работу и не закроет маркировку изготовителя электрорадиоэлемента, которая в любом случае должна быть отчетливо видна.

Форма паяных соединений - по возможности скелетная с вогнутыми галтелями припоя по шву и без его избытка. Она должна позволять визуально просматривать через тонкий слой припоя контуры входящих в соединение отдельных монтажных элементов. Поверхность галтелей припоя по всему периметру паяного шва - вогнутая, непрерывная, гладкая, глянцевая, без темных пятен и посторонних включений. Не допускается растекание припоя за пределы контактных площадок по проводнику, так как это уменьшает расстояние между соседними паяными соединениями или проводниками. Допускаются приемке без подпайки следующие паяные соединения печатного монтажа:

а) с заливной формой пайки, при которой контуры отдельных монтажных элементов, входящих в соединение, полностью скрыты под припоем со стороны пайки соединения;

б) с частичным незаполнением припоем металлизированных отверстий печатного монтажа, если высота заполнения составляет не менее 2/3 всей высоты отверстия;

в) с растеканием припоя по выводам, печатным проводникам и контактным площадкам с обеих сторон платы, если припой не затекает под корпус радиоэлементов, микросхем и не уменьшает минимально допустимого расстояния между соседними площадками или проводниками, оговоренного в чертеже;

г) с наличием отдельных мелких газовых или усадочных пор.

Качество паяных, подпаяных и перепаяных соединений контролируется визуально у 100% монтажных соединений. Контроль линейных величин допустимых дефектов производится с помощью любого мерительного инструмента, обеспечивающего требуемую точность. При контроле качества паяных, подпаяных иди перепаяных соединений допускается:

а) применять метод сравнения с эталонными образцами паяных соединений;

б) применять при визуальном осмотре паяных швов лупы, очки-в) по требованиям заказчика производить дополнительный выборочный контроль с целью выявления в соединениях скрытых дефектов с помощью рентгенотелевизионного микроскопа;

г) проверять механическую прочность паяных соединений при наличии в технических условиях на изделие требований к их прочности;

д) применять контроль сопротивления контактного перехода паяных соединений зондовым методом.

Паяные соединения на механическую прочность испытывают на разрывных машинах. Проверка проводится выборочно на контрольных образцах. Критерием оценки механической прочности является величина сопротивления срезу или отрыву паяного соединения, которая должна составлять не менее 0,5 кг. В отдельных случаях допускается проводить проверку механической прочности специальным пинцетом непосредственно на изделиях, при этом усилие должно быть направлено вдоль оси припаянного провода.

Одно из важнейших условий надежной и стабильной работы электронной аппаратуры с полупроводниковыми приборами - эффективная отдача ими тепла в окружающее пространство. Нормальная работа диодов и транзисторов малой мощности (до 200 мВт) автоматически обеспечивается, если их режимы и температура окружающей среды не превышают максимально допустимых для конкретных приборов значений. Эффективное использование транзисторов большой мощности (свыше 1 Вт) возможно при условии, что они смонтированы на радиаторах - металлических пластинах или металлических конструкциях иной формы, способствующих отдаче тепла от транзисторов в окружающую среду. При этом тепло от электронно-дырочного перехода передается корпусу транзистора, он отдает тепло радиатору, а последний рассеивает его в окружающее пространство. Эффективность отдачи транзистором тепла радиатору зависит от качества теплового контакта между транзистором и радиатором и оценивается по величине теплового сопротивления корпус транзистора - радиатор. Это сопротивление тем меньше, чем лучше обработаны прилегающие поверхности транзистора и радиатора. Заполнение зазора между транзистором и радиатором смолой, кремнийорганическим составом или густым невысыхающим маслом (например, силиконовым) снижает тепловое сопротивление контакта. Аналогичный эффект дают прокладки из свинцовой фольги. Интегральная микросхема (ИМС) - микроэлектронное изделие, выполняющее определенную функцию преобразования и обработки сигнала и имеющая высокую плотность упаковки электрически соединенных элементов, которые с точки зрения требования к испытаниям, приемке, поставке и эксплуатации рассматриваются как единое целое. Коммутация микросхемы осуществляется при помощи печатных плат. Формовка выводов микросхемы применяется для увеличения расстояния между выводами, совмещения выводов с отверстиями и контактными площадками печатной платы, фиксации расстояния от корпуса микросхемы до платы. Также применяется установка микросхем без формовки выводов. Интегральные схемы со штыревыми выводами устанавливают только с одной стороны печатной платы на расстоянии 1 .3 мм от монтажной плоскости до корпуса платы. Этот зазор необходим для устранения перегрева микросхемы при пайке и для возможности нанесения защитного покрытия. Для дополнительного механического крепления возможна установка ИМС на специальную подставку. Интегральные микросхемы в корпусах с планарными выводами приклеивают эпоксидным клеем с одной или двух сторон платы. Если под корпусом микросхемы проходят проводники, то его устанавливают на плату с зазором или на прокладку из стеклоткани толщиной 0,2…0,3 мм. Основные операции технологического процесса монтажа микросхем на печатной плате: входной контроль печатных плат и микросхем, формовка, обрезка и лужение выводов, установка микросхем на печатные платы, пайка электрических параметров, покрытие лаком, окончательный контроль.Групповые методы пайки обычно применяют при одностороннем расположении навесных деталей на печатной плате. При двустороннем монтаже групповая пайка возможна только с одной стороны, а пайка с другой стороны производится монтажным паяльником. Различают несколько типов групповой пайки: пайка погружением, пайка волной припоя, каскадная пайка, избирательная пайка. Технологический процесс пайки печатных плат с односторонним монтажом методом погружения и волной припоя состоит из следующих этапов: обезжиривание, наклейка маски, пайка, удаление маски и остатков флюса, контроль. Обезжиривание выполняют погружением платы со стороны монтажа в растворитель, состоящий из смеси спирта с бензином. Затем плату обдувают воздухом до полного высыхания. Участки и проводники платы, которые не подвергаются пайке, закрывают маской. Последнюю штампуют из бумажной ленты, покрытой костным клеем. В маске пробивают отверстия против мест пайки и базовые, маску приклеивают так, чтобы места пайки не выходили за пределы отверстий в маске.

При пайке погружением плату устанавливают в приспособление с вибрационной головкой и погружают на 4 6 см в расплавленный флюс (обычно 40% канифоли и 60 этилового спирта), а затем в припой (ПОС61), расплавленный при помощи нагревателя. Через 1 с после погружения включают вибратор, что создает условия для проникновения флюса и припоя в отверстия и способствует правильному оформлению пайки. Амплитуда вибрации устанавливается экспериментально для каждого типа и размера плат в пределах такого максимального значения, при котором не происходит разбрызгивания флюса и припоя. Время выдержки при температуре припоя 240°С составляет 6 11 с, а при температуре припоя 250°С 4 .8 с. По окончании пайки плату извлекают из припоя и, не включая вибратор, выдерживают над ванной 5-7 с.

К недостаткам пайки погружением относятся коробление плат вследствие температурных деформаций, необходимость поддержания постоянной высоты уровня припоя в ванне и быстрое окисление расплавленного припоя.

Пайка волной припоя не имеет этих недостатков. В ванне находится припой, расплавленный нагревателем. Печатная плата проходит по гребню волны, которая создается подачей припоя через сопло крыльчаткой. Контакт платы с постоянным притоком припоя обеспечивает быструю передачу тепла, что сокращает время пайки.

Для удаления маски плату погружают на 0,8 .0,9 ее толщины в ванну с горячей (90 С) водой и выдерживают до тех пop пока она не отклеится (2 3 мин). Затем плату обдувают горячим воздухом до полного высыхания.

Удаление флюса осуществляется погружением в ванну со смесью бензина (50%) и спирта (50%) на 2-4 мин.

* 1. **Практическое задание**

В рамках практики была разработана программа для микроконтроллеров Arduino, а так же программа на языке Python для коммуникации с микроконтроллером через COM порт и графического отображения данных, полученных с контроллера.

Ниже представлены листинги программ на Python и C++, написанные для работы с контроллером Arduino.

1. Коммуникация с контроллером.

import serial #requires pip install pyserial

import matplotlib.pyplot as plt

import numpy  as np

serial\_port = 'COM19' #set your COM port value

baud\_rate = 9600; #set same as Arduino has

write\_to\_file\_path = "output.txt"

with serial.Serial(serial\_port, baud\_rate, timeout = 0) as ardu:

    with open(write\_to\_file\_path, "w") as output\_file:

        num = input("Enter a number: ") # Taking input from user

        ardu.write(bytes(num, 'utf-8')) #Input -> COM port

        while True:

            # time.sleep(0.01)

            value = ''

            while (value.find('\n') == -1):

                value += ardu.readline().decode('utf-8')

            value = value.strip()

            if (value == 'END'):

                break

            print(value) # printing the value

            output\_file.write(value + '\n')

data = np.loadtxt('output.txt')

y = data[:]

plt.plot(y)

plt.ylabel('y')

plt.show()

Программа приглашает пользователя ввести числовую команду, затем отправляет команду в выбранный COM порт и ждет поступления данных в порт от контроллера. Затем записывает данные в файл и на основе данных из файла строит график.

2. Алгоритм работы микроконтроллера.

Работа платы полностью описывается одним классом.

Класс — в объектно-ориентированном программировании, представляет собой шаблон для создания объектов, обеспечивающий начальные значения состояний: инициализация полей-переменных и реализация поведения функций или методов.

Классовая структура программы позволяет быстро перенастраивать программу для работы с различными разрядностями шин, а так же менять места их подключения на плате.

Диаграмма класса на языке UML представлена на рис. 6.

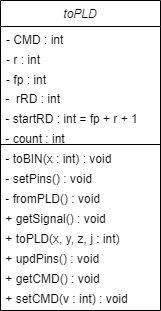


Рис. 6.

Листинг программы:

#include <Arduino.h>

int s;

double pi = 3.1415;

class toPLD

{

private:

  int CMD; //Содержит введенную команду

  int r;   //Разрядность шины

  int fp;

  int rRD; //Hазрядность принимаемых данных

  int startRD = fp + (r + 1);

  int count;        //Количество отсчетов сигнала

  void toBIN(int x)

  {

    for (int i = r - 1; i >= 0; --i)

    {

      int f = ((int)((x >> i) & 1));

      switch (f)

      {

      case 1:

        digitalWrite(i + fp, HIGH);

        break;

      case 0:

        digitalWrite(i + fp, LOW);

        break;

      }

    }

    digitalWrite(r + fp, HIGH);

    delay(1000);

    digitalWrite(r + fp, LOW);

  }

  void setPins()

  {

    for (int i = fp; i <= r + fp; i++)

    {

      pinMode(i, OUTPUT);

    }

    for (int i = r + fp + 1; i <= r + fp + rRD; i++)

    {

      pinMode(i, INPUT);

    }

  }

  void fromPLD()

  {

    String s = "";

    for (int i = fp + (r + 1); i <= fp + r + rRD; i++)

    {

      int c = digitalRead(i);

      s += (char)('0' + c);

    }

    int result = 0;

    for (int i = s.length() - 1, pos = 0; i >= 0; i--, ++pos)

      result += s[i] == '0' ? 0 : 1 << pos;

    Serial.println(result);

  }

public:

  void getSignal() //Получаем числа с шины данных и выводим в порт

  {

    for (int i = 0; i < count; i++)

    {

      digitalWrite(fp + r + rRD + 1, HIGH);

      fromPLD();

      digitalWrite(fp + r + rRD + 1, LOW);

    }

    Serial.println("END");

  }

  toPLD(int x, int y, int z, int j) //Конструктор класса

  {

    r = x;     //Разрядность шины управления

    fp = y;

    rRD = z;   //Разрядность шины данных

    count = j; //Количество отсчетов сигнала с плис

  }

  void updPins()

  {

    setPins();

    if (CMD != -1)

      toBIN(CMD);

  }

  int getCMD() //Просто так

  {

    return CMD;

  }

  void setCMD(int v)

  {

    if ((v >= 0) && (v <= pow(2, r) - 1))

      CMD = v;

    else

    {

      Serial.println("Ардуинка не понимает такого обращения");

      CMD = -1;

    }

  }

};

void setup()

{

  Serial.begin(9600);

}

void loop()

{

  toPLD arduino(2, 2, 6, 16384);

  while (true)

  {

    if (Serial.available())

    {

      String ss = Serial.readStringUntil('\n');

      if (ss != "")

      {

        int s = atoi(ss.c\_str());

        arduino.setCMD(s);

        arduino.updPins(); //Обновляем управляющий сигнал

        delay(500);

        pinMode(4, INPUT);

        pinMode(13, INPUT);

        while (digitalRead(4) != 1)

        {

        }

        arduino.getSignal(); //Получаем отсчеты сигнала с ПЛИС

      }

    }

  }

}

Конструктор класса принимает значения: разрядности управляющей шины, номер первого пина, разрядность шины данных, подключенной к выходу ПЛИС, а так же количество остчетов сигнала, приходящих с ПЛИС.

 toPLD(int x, int y, int z, int j) //Конструктор класса

  {

    r = x;     //Разрядность шины управления

    fp = y;    //Первый пин. Порядок: пины управления -> пин RDY -> шина приема данных -> пин RDY для шины данных

    rRD = z;   //Разрядность шины данных

    count = j; //Количество отсчетов сигнала с плис

  }

Это позволяет быстро перенастроить программу для работы с другими микроконтролерами или же в других условиях, например при изменении разрядности шин.

Метод безопасного приема числового значения команды внутрь класса не позволяет ввести команду, значение которой превысило бы разрядность управляющей шины.

  void setCMD(int v)

  {

    if ((v >= 0) && (v <= pow(2, r) - 1))

      CMD = v;

    else

    {

      Serial.println("Ардуинка не понимает такого обращения"); //Исключение, ничего не выполняется

      CMD = -1;

    }

  }

};

Публичный метод updPins() вызывает приватные методы, которые перепределяют входные и выходные пины платы, а затем передают команду на управляющую шину.

  void updPins()

  {

    setPins();

    if (CMD != -1)

      toBIN(CMD);

  }

  void toBIN(int x)

  {

    for (int i = r - 1; i >= 0; --i)

    {

      int f = ((int)((x >> i) & 1)); //преобразование куска числа в двоичный код, начиная со старшего разряда

      switch (f)

      {

      case 1:

        digitalWrite(i + fp, HIGH);

        break;

      case 0:

        digitalWrite(i + fp, LOW);

        break;

      }

    }

    digitalWrite(r + fp, HIGH); //Импульс READY, означает окончание формирования сигнала на шине

    delay(1000);

    digitalWrite(r + fp, LOW);

  }

  void setPins() //Инициализирует выходные пины в зависимости от разрядности шины

  {

    for (int i = fp; i <= r + fp; i++)

    {

      pinMode(i, OUTPUT);

    }

    for (int i = r + fp + 1; i <= r + fp + rRD; i++)

    {

      pinMode(i, INPUT);

    }

  }

Далее вызывается метод getSignal(). Он последовательно считывает 8-разрядные числа с ПЛИС и выводит их в COM порт. Программа на Python записывает эти значения в файл и строит из них график.

  void getSignal() //Получаем числа с шины данных и выводим в порт

  {

    for (int i = 0; i < count; i++)

    {

      digitalWrite(fp + r + rRD + 1, HIGH);

      fromPLD();

      digitalWrite(fp + r + rRD + 1, LOW);

    }

    Serial.println("END"); //Для завершения цикла записи в файл в Python

  }

  void fromPLD() //Принимает код с шины ПЛИС, преобразует в десятичное число, выводит в порт

  {

    String s = "";

    for (int i = fp + (r + 1); i <= fp + r + rRD; i++)

    {

      int c = digitalRead(i);

      s += (char)('0' + c);

    }

    int result = 0;

    for (int i = s.length() - 1, pos = 0; i >= 0; i--, ++pos)

      result += s[i] == '0' ? 0 : 1 << pos;

    Serial.println(result);

  }

После этого цикл работы программы заканчивается и начинается новый.