Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Радиофизический факультет
Кафедра радиоэлектроники

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ООП
к.ф.-м.н., доцент
_____В.А. Мещеряков
«_15_» февраля 2018 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА СПЕЦИАЛИСТА

РАЗРАБОТКА И ИЗГОТОВЛЕНИЕ МАКЕТА ПУЛЬТА ДИСТАНЦИОННОГО УПРАВЛЕНИЯ НА БАЗЕ ATMEGA8

по основной образовательной программе подготовки специалиста
по специальности 11.05.01 «Радиоэлектронные системы и комплексы»

Репин Александр Андреевич

Руководитель ВКР
кандидат физ.-мат. наук, доцент
_____Мещеряков В.А.
«_13_» февраля 2019 г.

Автор отчета
студент группы № 738
_____Репин А.А.
«_13_» февраля 2019 г.

Томск - 2019

Министерство образования и науки Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Радиофизический факультет
Кафедра радиоэлектроники

УТВЕРЖДАЮ
Руководитель ООП
к.ф.-м.н. доцент
_____ В.А. Мещеряков
«28» декабря 2018г.

ЗАДАНИЕ
на подготовку выпускной квалификационной работы (ВКР)
студенту 738 группы Репину Александру Андреевичу

1. Тема ВКР: Разработка и изготовление макета дистанционного пульта управления на базе микроконтроллера ATMEGA8.
2. Срок сдачи студентом выполненной ВКР:
   а) на кафедре 10.02.2019,
   б) в ГЭК 14.02.2019
3. Краткое содержание работы:
Задание ориентировано на написание выпускной квалификационной работы по теме Разработка и изготовление макета дистанционного пульта управления на базе микроконтроллера ATMEGA8.

4. Календарный график выполнения работы:

а) оформление литературного обзора	15.01.2019–22.01-2019
б) оформление материалов по устройству управления датчиком	15.01.2019–22.01-2019
в) оформление конструкторско-технологической документации	22.01.2019–29.01-2019
г) оформление выводов по работе	29.01.2019–05.02-2019
д) техническое оформление ВКР в соответствии с требованиями	29.01.2019–05.02-2019
е) оформление документов на цитируемую чистоту работы	05.02.2019–12.02-2019
ж) оформление документов для сдачи работы в библиотеку	05.02.2019–12.02-2019

5. Дата выдачи задания «28» ___декабря_____2018 г.

Руководитель ВКР –
кандидат физ.-мат. наук,
доцент кафедры радиоэлектроники _____ Мещеряков В.А.

Задание принял к исполнению _____ Репин А. А.

РЕФЕРАТ

Выпускная квалификационная работа с. 62, гл. 6, рис. 20, табл.3, источников 25, приложений 6.

ПДУ, МИКРОКОНТРОЛЛЕР, СВЕТОДИОД, ИК-ИЗЛУЧЕНИЕ, ПЕРЕДАТЧИК, ПРИЕМНИК

Цель работы: разработка и изготовление макета пульта дистанционного управления (ПДУ) по инфракрасному каналу связи.

В данной работе разработана и изготовлена схема пульта дистанционного управления (ПДУ) по инфракрасному каналу связи.

Проведен анализ выбора составляющих частей схемы, а именно выбор светодиода, фотодиода, клавиатуры, микроконтроллера, индикатора. Рассмотрен вопрос передачи и приема по инфракрасному каналу связи, а также программирование микроконтроллеров. Представлена программа для микроконтроллера, которая написана на языке ASSEMBLER для передающей и приемной частей ПДУ. Приведена принципиальная схема устройства ПДУ.

В ходе работе были изучены характеристики и дано краткое описание микроконтроллера Atmega8535 и Atmega8 семейства AVR, построенной на базе прогрессивной архитектуры RISC с применением программируемой памяти EPROM.

В работе представлена схема ПДУ, предоставлено описание построения программы.

В результате работы:

1. изучены принципы программирования;
2. написана программа передающей и приемной частей ПДУ;
3. разработан макет передающей и приемной частей ПДУ.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 4

1 Литературный обзор. Пульт дистанционного управления 6

2 Выбор компонентов ПДУ 11

2.1 Клавиатура 11

2.2 Светодиод инфракрасного излучения 14

2.3 Инфракрасный фотодиод 18

2.4 Светодиодный индикатор 19

2.5 Усилитель Дарлингтона 20

2.6 Цифровые микроконтроллеры 21

3 Схемная реализация передающей части ПДУ 26

4 Программное обеспечение для передающей части ПДУ 28

5 Схемная реализация приемной части ПДУ 31

6 Программное обеспечение приемной части ПДУ 33

ЗАКЛЮЧЕНИЕ 35

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 36

Приложение А (справочное) Текст программы для передающей части ПДУ 38

Приложение Б (справочное) Текст программы для приемной части ПДУ 44

Приложение В (справочное) Характеристики микроконтроллера Atmega8 55

Приложение Г (справочное) Характеристики микроконтроллера Atmega8535 56

Приложение Д (справочное) Конструкторская документация передающей части ПДУ 57

Приложение Е (справочное) Конструкторская документация приемной части ПДУ 60

ВВЕДЕНИЕ

Промышленные технологии развиваются семимильными шагами, и цифровая электроника не исключение. Сейчас широко используется устройства с цифровой логикой функционирования. Цифровые устройства – это устройства, которые в большинстве случаев, для обработки информации используют цифровые логические схемы [1]. Различие между цифровыми и аналоговыми схемами заключаются в том, что цифровые схемы работают с дискретным сигналом, а аналоговые с непрерывными. Цифровые сигналы – это сигналы, информационные параметры (частота, фаза, амплитуда) которых можно представить в виде последовательности дискретных (цифровых) значений [2]. Цифровые технологии шагнули далеко вперед, ведь в современном мире почти у каждого человека дома есть ноутбуки, смартфоны, плееры, домашние кинотеатры, для управления которыми достаточно нажатия на кнопку компьютерной мыши или пульта.

Технических прогресс не стоит на месте, и не перестают удивлять нас новинками. К примеру, еще каких-то 50 лет назад, компьютеры имели очень большие габариты, занимали много места, функционал и быстродействие которых оставляли желать лучшего. На сегодняшний день, есть такие компьютеры, которые очень малогабаритны, а их быстродействие позволяет им выполнять операции буквально за доли секунд. Также цифровые устройства имеют ряд преимуществ перед аналоговыми, а именно передача сигнала без искажений, хранение и извлечение данных без искажений, и управление программным обеспечением без замены аппаратных средств. Основными устройствами, которые отвечают за управление в цифровой технике являются микроконтроллеры и микропроцессоры.

Микроконтроллер – это микросхема, которая предназначена для управления электронными устройствами [3]. Микросхема, которая программно-управляемая, и способна выполнять функции полноценного компьютера. К микроконтроллерам можно подключать периферийные устройства (клавиатура, светодиодные панели, разные двигатели), которые управляются через порты ввода/вывода. Широкое применение нашли микроконтроллеры AVR фирмы «Atmel», семейства Mega, который является 8-разрядным и предназначен для встраивания в различные приборы [4].

Управление устройствами и приборами на расстоянии с помощью пультов дистанционного управления (ПДУ) – актуальная тема. ПДУ нашли широкое применение во многих областях жизни человека. Например, они применяются для управления аппаратами и механизмами на мобильных объектах, таких как самолеты, космические корабли, корабли, также ПДУ актуальны и для управления производственными процессами, системами связи, техникой повышенной опасности. Пульты не обошли стороной и повседневную жизнь человека и применяются в бытовой технике, например, телевизоры, машинные сигнализации,

чайники. Но есть конечно же и недостатки, такие как высокая цена и ограниченность функций, что позволяет пользоваться ПДУ не всем.

Цель данной работы заключается в разработки и изготовлении макета приемной части пульта дистанционного управления, который позволит изменять его характеристики, будет иметь широкий спектр применения и конечно же стоимость которого будет невысока. Также данная работа может использоваться в образовательных целях, для изучения курса «Микроконтроллеры» в качестве готового проекта, примера как применение микроконтроллеров. Пульт дистанционного управления будет выполнен на базе микроконтроллеров семейства Mega, между приемной и передающей частями данные передаются по инфракрасному каналу связи.

1 Литературный обзор. Пульт дистанционного управления

Пульт дистанционного управления (ПДУ) – это электронное устройство для удаленного (дистанционного) управления устройством на расстоянии [5]. ПДУ применяют в различных сферах, такие как управление бытовой техникой, военной техникой и устройствами тяжелой промышленности. Яркий пример использования ПДУ – с помощью ПДУ управление бытовой техникой: телевизором, домашним кинотеатром и т.д. Военные используют ПДУ для управления беспилотными летательными аппаратами и баллистическими ракетами. В промышленности – управление механизмами на мобильных объектах (корабли, самолеты, космические корабли). Ниже на рисунке 1 изображены примеры использования ПДУ.

Рисунок 1 – Пульты дистанционного управления

Под цифрой 1 – стационарный пульт командного пункта, для управления запуском ракет (СССР). Под цифрой 2 – пульт для управления краном, подъемником. 3 – пульт для управления беспилотными летательными аппаратами. Под 4 цифрой – пульт для управления бытовой электроникой, телевизор видеоплеер и т.д.

Существует классификация ПДУ по различным параметрам, которые и будут описаны.

В зависимости от питания бывает двух типов. Самый распространенный тип – это автономное питание. Ко второму типу относится неавтономное, то есть питание по кабелю [5].

Пульты дистанционного управления различаются по каналам связи, таким как: проводной канал, радиоканал, ультразвуковой канал, инфракрасный канал, каждый из которых рассмотрим более подробно [5].

Предназначение проводного канала связи заключается в передаче сигналов от одного устройства к другому, который находится на расстоянии [5]. Проводной канал применяется в тех случаях, когда невозможно применить беспроводные каналы связи. На это могут быть разные причины, например, такие как отсутствие прямой видимости, наличие экранировки, соображений маскировки, помехоустойчивости. Подобный принцип работы у кабельного телевидения, где из-за экранировки сигнала домами, целесообразно использовать проводной канал. Также более эффективно использование кабельного канала в управлении мобильных, военных объектов, лабораторий и оборудования производственных объектов.

Радиоканал – канал связи, в котором передача информации осуществляется с помощью радиоволн [6]. Радиоканал используют для объектов, которые находятся в движении, беспилотных летательных объектов, в чрезвычайных ситуациях для управления роботами и т.д. Также радиоканал используется в тех случаях, когда приемник и передатчик не могут находится в зоне прямой видимости.

Ультразвуковой канал менее распространен и используется редко, для управления стационарными объектами на небольшом расстоянии [7].

Инфракрасный канал связи не требует соединительных проводов, потому что используется для связи инфракрасное излучение. Главное его преимущество по сравнению с радиоканалом – нечувствительность к электромагнитным препятствиям [8]. Инфракрасный канал используется в бытовой электронике. Чаще всего в телевизорах, аудио и видеотехнике.

Еще один параметр по которому разделяют ПДУ – это мобильность [5]. По этому параметру разделяют два вида ПДУ: стационарный и носимый. Стационарный тип чаще всего используется в военной технике, различного рода лабораториях и в автоматизированных заводах. Носимый тип для управления мобильными устройствами или бытовой техникой.

Также пульты различают по функционалу: обучаемые, универсальные и с фиксированным набором команд [5]. В обучаемых пультах имеются возможность перепрограммирования, что нельзя сказать о остальные типы ПДУ. Особенностью универсальных пультов является то, что он включает в себя разный набор сигналов для управления различными устройствами. Пульты с фиксированным набором команд – это заводские пульты, выполняющие только заводскую программу.

Так как в моей работе ПДУ будет разработан с ИК каналом связи, то рассмотрим подробнее пульты для управления бытовой электроникой.

Чаще всего бытовые ПДУ питаются от 2 – 4 батареек типоразмера АА или ААА, иногда от батарейки 9 В типа «Крона» [4]. Это связано с тем, что для питания инфракрасного светодиода необходимо не менее 2,0 – 2,5 Вольта и от одной батарейки 1,5 В (без использования изощрённых схемотехнических приёмов) такого напряжения не получить.

Чаще всего, для бытовой электронной аппаратуры пульт дистанционного управления – небольшое устройство с кнопками и питаем от батареек, которое посылает информационный сигнал по каналу инфракрасного излучения с длиной волны 0,75–1.4 микрон [10]. Свет, который излучается из передатчика нельзя увидеть человеческим глазом, а распознается только приемником принимающего устройства. Распространенный вариант ПДУ применяется специализированная микросхема, которая находится в корпусе, либо без использования корпуса. Данная микросхема устанавливается на печатную плату и заполняется специальной термопластической полимерной смолой, которая называется компаундом, позволяющая защищать микросхему от повреждений. На рисунке 2 представлена схема системы управления объектами через ИК канал.

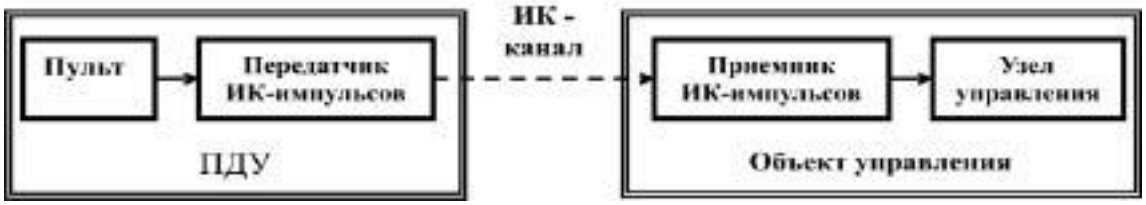

Рисунок 2 – Схема системы управления объектами с помощью ИК излучения

Те пульты, которые имеют несколько функций применяется сложная система управления. Эта система с частотной модуляцией несущего сигнала, которая применяется в тех случаях, когда необходима помехозащищенность канала и кодирование передаваемых команд [10]. Для цифровой обработки используется микроконтроллер передатчика, находящегося в пульте, который модулирует и кодирует передаваемый сигнал, соответственно в приемнике осуществляется демодуляция и декодирование [10]. Поле того как завершается процесс демодуляции, применяются соответствующие частотные фильтры, назначение которых разделение сигналов.

В компьютерных клавиатурах для считывания кода при нажатии кнопки в большинстве случаев применяется метод сканирования матриц кнопок, но для ПДУ бытовой техники данный метод неуместен, потому что данный метод требует колоссальных энергетических затрат, что приведет к довольно быстрому разряжению батарейки. Из-за чего применяется метод «одна кнопка – одна линия» [11]. Суть метода заключается в том, что при нажатии на кнопку пульта соединяются общая линия и линия этой кнопки, после чего сигнал отправляется на микроконтроллер пульта, этот сигнал содержит код пульта и соответственно код кнопки.

У таких пультов отсутствует обратная связь, т.е. отсутствует информация о том, достиг или не достиг сигнал приемника. Из-за этого, сигнал, который соответствует нажатой кнопке, без перерыва передает информацию до того момента, пока кнопка не будет отпущена. В тот

момент, когда кнопка опустится пульт переходит в дежурное состояние.

Приемник получает данные, а именно проверяется код пульта, в том случае, если данный код соответствует этому устройству, то тогда исполняется команда, которая соответствует нажатой кнопке. Логично, что как приемник, так и передатчик должны использовать одинаковые методы кодирования и частоты модуляции передаваемых данных, иначе приемная сторона не имеет возможности принять и обработать данные, которые ей посылают.

Как правило, в пультах применяется одна частота модуляции, несущей излучения ИК светодиода, т.е. как пульт, так и приемник запрограммирован на эту частоту. Такие фирмы как Panasonic и Sony, чаще всего используют стандартные частоты, значения которых равны 36 кГц, 38 кГц и 40 кГц [10]. Фирмы Sharp использует более редкую частоту 56 кГц, а фирма Bang & Olufsen 455 кГц, что является также является достаточно редкой частотой [10]. В том случае, когда используемая приемником частотная модуляция не точно совпадает с частотной модуляцией передатчика, есть вероятность снижения чувствительности, но в тоже время сигнал приниматься будет [10]. Каждой кнопке пульта соответствует своя модуляция инфракрасного светодиода.

Инфракрасный светодиод передает сигнал соответствующей модуляции. Для частот от 30 до 50 кГц подходят светодиоды длина волны которых составляет 950 нм, а для частоты 455 кГц ориентированы светодиоды с длиной волны 870 нм. Соответственно, для этих специальных светодиодов с высокой частотной модуляцией используются специальные приемники TSOP5700 и TSOP7000 [9].

Фотоприемник состоит из нескольких каскадов усилителей и частотного детектора, чувствительность которого равна до -90 дБ. К примеру, большое количество радиолюбительских схем имеют чувствительность до – 60 дБ. Также следует отметить, что в составе всех ИК приемников, которые производятся серийно, имеется ИК – светофильтр, который представляет собой темно-красную линзу или пластинку. В априори модуль ИК приемника имеет три выхода: питание, земля, вывод данных. К примеру, фотоприемники, которые производит Vishay Telefunken : TSOP1736 — настроен на частоту 36 кГц, TSOP1738 — 38 кГц [10].

Для того, чтобы распознать большое количество команд используют кодирование данных. На сегодняшний день в основном используют две схемы кодирования передаваемых данных.

Первая схема носит название Манчестерское кодирование, для которой характерна использование протоколов RC4 и RC5, данную схему применяет фирма Philips [10]. Отличительной чертой такого кодирования является передача 0 дополняется 1,

соответственно 1 дополняется 0. Например, 001 передается ка 01 01 10. Из этого следует, что информационный сигнал считывается последовательно, и в эфир попадает модулированный сигнал, только в том случае, когда встречается единица.

Следующая схема применяется фирмой Sony. Ее суть в следующем, в модулированном вначале всегда передается «1», затем «0» – пауза [10]. Таким образом, временной размер единицы всегда остается одинаков, а временной размер нуля является кодированными передаваемыми данными. Соответственно, длинная пауза – передача единицы, короткая – передача нуля.

Перед посылкой кодированных данных пульт всегда посылает одну или несколько синхропосылок для того, чтобы фотоприёмник настроил приёмную цепь, тем самым синхронизировался с пультом по чувствительности и фазе [10].

2 Выбор компонентов передающей части ПДУ

2.1 Клавиатура

Клавиатура – периферийное устройство, представляющее собой совокупность кнопок, предназначенных для управления другим устройством или ввода в него информации. В данной работе с клавиатуры подается сигнал в МК. Каждой клавише присвоена своя подпрограмма, которая срабатывает при нажатии. Рассмотрим этот процесс подробнее. Начнем с того, что в нашем пульте используется клавиатура AK-304FM-N-WWB с 12 кнопками и семью ножками вывода. На рисунке 3 изображена данная клавиатура.

Рисунок 3 – Клавиатура AK-304FM-N-WWB

Как видно на рисунке, это готовое устройство для монтажа в схему или макетную плату. Сама схема защищена пластиковым корпусом, все цифры имеют свое обозначение. У данной клавиатуры 7 ножек, они расположены с обратной стороны устройства. В данной клавиатуре используется система опроса вертикальных и горизонтальных магистралей. То есть на первую горизонтальную магистраль посажены цифры 1, 2, 3 на вторую 4,5,6, третью 7,8,9 и четвертую*,0, ♯. Первая вертикальная магистраль соединяется с горизонтальными, кнопками 1,4,7,*, вторая 2,5,8,0, а третья 3,6,9 и ♯. Кнопки — это своего рода ключи, которые замыкают в линию горизонтальный и вертикальные магистрали [11]. Как видно на рисунке 7 из-под клавиатуры выходит 7 проводов, они соединены с ее ножками вывода. Первые четыре провода специально обозначены оранжевым цветом, это горизонтальные магистрали, три белые это вертикальные. В передатчике будут использоваться только девять кнопок, отсюда

следует, что одна ножка подсоединена к порту, но не активна и находится в резервном состоянии. Так что задействованных портов в микроконтроллере всего шесть штук, три на ввод и три на вывод. На рисунке 4 изображена схема клавиатуры.

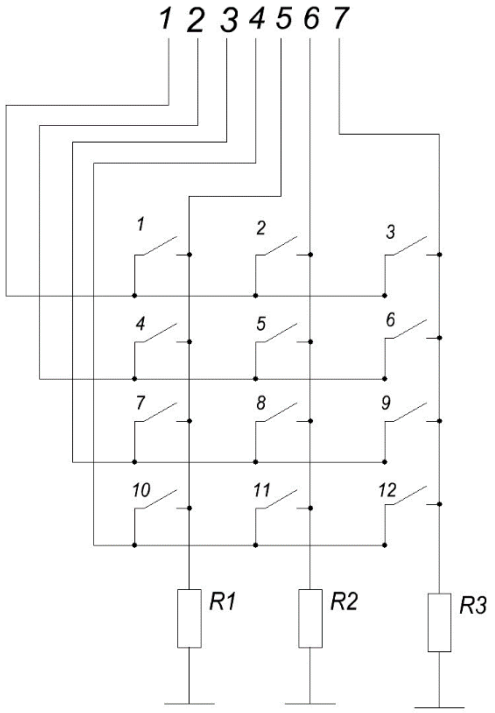

Рисунок 4 – Схема клавиатуры

На данной схеме видно, какие ножки вывода клавиатуры за что отвечают. Еще раз напомним, что ножка под номером 1 отвечает за первый горизонтальный ряд кнопок, 2 ножка за второй горизонтальный ряд, 3 за третий, 4 соответственно за четвертый, 5 ножка вывода — это первый вертикальный ряд кнопок, 6 – второй вертикальный ряд, 7 – соответственно третий ряд кнопок. В работе будут функционировать только девять кнопок, номиналом от 1 до 9, тем самым 4 ножка клавиатуры, отвечающая за нижний ряд кнопок, будет находиться, в резерве или другими словами отключена. Так как она имеет связь с микроконтроллером то в случае необходимости, с помощью изменения программного кода, ее можно будет задействовать.

Клавиатура сама по себе представляет набор ключей, которые в случае нажатия объединяют вертикальные и горизонтальные магистрали в линию, а микроконтроллер — это микросхема, которая сканирует данные магистрали в поисках нажатой кнопки или ключа. Горизонтальные магистрали клавиатуры подключены к ножкам ввода/вывода микроконтроллера. Первые четыре ножки клавиатуры по порядку соответствуют портам PD0, PD1, PD2, PD3. С пятой по седьмую ножку так же по порядку соответствуют портам PD4, PD5, PD6. На рисунке 5 изображена схема подключения клавиатуры к микроконтроллеру.

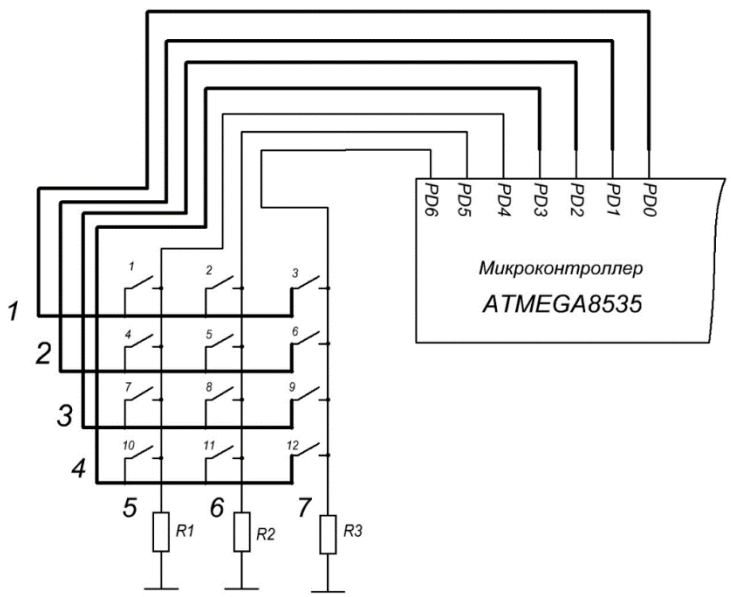

Рисунок 5 – Схема подключения клавиатуры к микроконтроллеру

В данной схеме используется динамический опрос клавиатуры. Начнем с того что порты микроконтроллера PD0, PD1, PD2, PD3 настроены на вывод информации, или другими словами выдачу напряжения, а порты ввода PD4, PD5, PD6 на прием информации. Подавая напряжение на порт вывода PD0, подается сигнал на первую горизонтальную магистраль клавиатуры, далее переходим на опрос вертикальных магистралей, при нажатой кнопке первого горизонтального ряда, на одном из портов приема появится сигнал. Таким образом, высчитывается номер нажатой клавиши. Если же на вертикальных портах ничего не появилось, то на PD0 прерывается подача сигнала, а на последующем порте вывода подается сигнал, после опрос портов приема снова повторяется и т.д. Таким образом, создается цикл опроса. При нажатии кнопки микроконтроллер переходит на выполнение программы заданной для данной клавиши. После ее выполнения переходит снова в цикл опроса клавиатуры.

До того как в данной работе была использованна данная клавиатура, рассматривались различные варианты реализации управления пультом. В одной из них нужно было каждой клавише присвоить свой порт и запрограммировать его на ввод, дело в том что это действительно проще, но в микроконтроллере портов ограниченное количество и в целях экономии мы сделали клавиатуру с задействованным меньшим их количеством. При этом реализация данного варианта была бы неудобна для пользования, так как кнопки которые были приобретены неустойчиво держались в плате и занимали бы больше места. Во втором варианте использовалась схема аналогичная данной клавиатуре, она занимала много места на плате при этом остались проблемы с кнопками которые то и дело отлетали, не говоря уже о

виде самого устройства. Таким образом данная клавиатура нам идеально подошла, начиная с удобства реализации программы, стоимости и надежности.

2.2 Светодиод инфракрассного излучения

В ходе данной работы для связи приемника и передатчика для пульта дистанционного управления используется инфракрасный канал связи.

Для передатчика был приобретен светодиод номиналом L-34F3C работающий в инфракрассном спектре, аргументы в пользу выбора будут представлены ниже. Данный светодиод является маломощным, ток питания для него равен 20 мА и прямое напряжение равно 1.2, что не превышает максимальное значения тока питания порта микроконтроллера которое составляет 40 мА и напряжения 5В. Для его подключения достаточно ножку анода подключить к выводу порта, а катод через сопротивление на землю. Номинал сопротивления для подключения к порту был вычислен по закону Ома

$$I = \frac{U}{R}.$$

Так как подаваемое напряжение с порта равно 5В а потребляемый ток 20 мА, то получилось следующее

$$R = \frac{U}{I} = \frac{5}{20*10^{-3}} = 250 \text{ Ом}.$$

Так как 250 Ом не является стандартным сопротивлением было использовано сопротивление номиналом 220 Ом. Со схемой подключения можно ознакомится выше на рисунке 4. Факт того что светодиод маломощный облегчает схемотехническую реализацию устройства, так как большенство ИК светодиодов являются мощными и для них приходится использовать транзистор в ключевом режиме другими словами буферный эллемент. Подключение мощных светодиодов рассматривалось ниже на рисунке 6.

На рисунке 6 изображен ИК светодиод L-34F3C.

Рисунок 6 – Инфракрасный светодиод L-34F3C

Диаметр данного светодиода составляет 3мм, это меньше чем у других рассмотренных нами моделях таких как L-53P3C, FYL-50131RAB, АЛ107А, BL-L513IRAB диаметр которых составляет 5мм и ток питания около 100 мА, что относит их к типу мощных светодиодов. Отличается он не только размерами и питанием, но и стоимостью. Цена на данный светодиод в два раза меньше чем у вышеперечисленных светодиодов. Длина волны светодиода L-34F3C равна 940 нм [10], что идеально входит в спектр чувствительности фотоприемника. В связи с тем, что длина волны светодиода L-34SF4C равна 880 нм, а фотоприемник выбран с длиной волны 940 нм, то выбран светодиод L-34F3C. В таблице 1 представлены сравнение технических характеристик использованного в данной работе светодиода инфракрасного излучения L-34F3C и L-34SF4C.

Таблица 1 – Сравнение технических параметров ИК светодиода L-34F3C и L-34SF4C

Название характеристики	Показатели светодиода L-34F3C	Показатели светодиода L-34SF4C
Мощность излучения P, мВт	20	20
Прямое напряжение, В	1,2	1,3
При токе Iпр, мА	20	20
Длина волны, нм	940	880
Ширина спектра излучения, нм	50	50
Видимый телесный угол, град	30	30
Максимальное обратное напряжение, В	5	5
Максимальный прямой ток, мА	50	50
Максимальный импульсный прямой ток, мА	1200	1200
Способ монтажа	В отверстие	В отверстие
Рабочая температура, С	-40..85	-40..85
Цена, рубли	15	16

Подключение светодиодов к микроконтроллеру.

Для связи микроконтроллера с переферийными устройствами существуют порты ввода/вывода, к которым и в том числе подключается светодиоды. Для микроконтроллеров серии AVR существует ограничение, максимальны ток, который может перенести порт ввода/вывода равен 40 мА. В связи с чем, рабочее значение тока вывода линии порта составляет 20 мА и не может быть больше 30 мА.

По этому рабочее значения тока вывода линии порта составляет 20 мА и не может превышать 30 мА. В зависимости от тока потребления светодиода в рабочем режиме подключение может различаться. Если ток потребления светодиода не превышает 30 мА, а номинальное напряжение питание не превышает напряжение питания микроконтроллера, то его можно непосредственно подключить к линии порта ввода/вывода. При этом порт должен быть запрограммирован на режим вывода. На рисунке 7 изображенно подключение маломощного светодиода к микроконтроллеру.

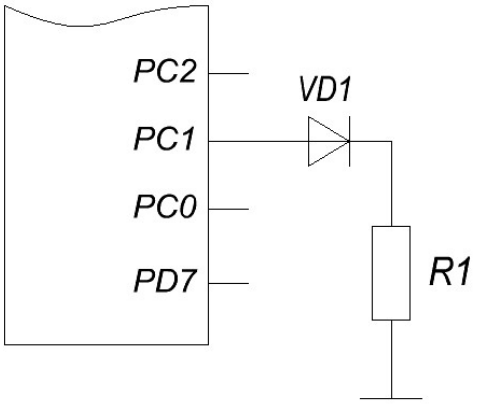

Рисунок 7 – Подключение светодиода к порту ввода/вывода

При подключении более мощных светодиодов, ток питания которых, не удовлетворяет предельному значению тока линии порта вывода, или напряжение питание больше чем значение питание микроконтроллера, тогда для подключения необходимо использовать буферный элемент, то есть транзистор в ключевом режиме. На рисунке 8 изображено подключение мощного светодиода к МК.

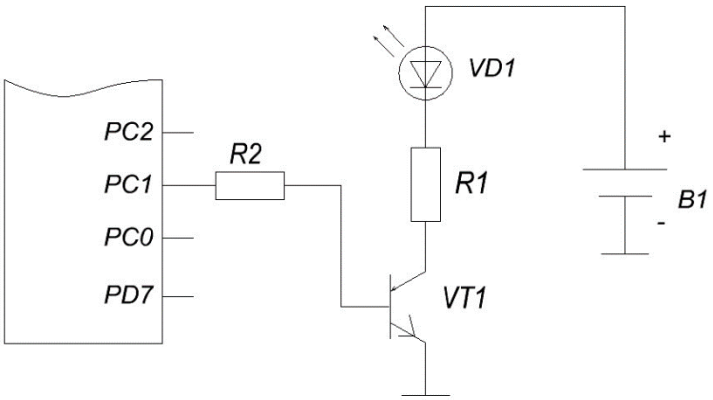

Рисунок 8 – Схема подключения мощного светодиода к МК

При наличии логического нуля на PC1 транзистор VT1 закрыт и ток не течет через светодиод. Установив в порте PC1 логическую единицу, открывается транзистор VT1, который подключает катод светодиода к минусу питания, подавая на него напряжение. Таким образом, решилась проблема токовых ограничений вывода порта. В данном случае необходимо учитывать, что значение максимального допустимого тока коллектора транзистора должно быть больше чем значение номинального светодиода.

В пульте дистанционного управления, будет использованы два типа передачи сигнала с разной длительностью и частотным разделением. На передатчике и приемнике установлены тумблеры переключения режимов.

Частотная передача представлена без кодировки сигнала. Тем самым каждой клавише в зависимости от ее номинала присвоена своя частота зажиганий и гашений, другими словами миганий. Одно мигание занимает около одной четвертой секунды, за которое светодиод успел зажечься и потухнуть. Это было сделано для визуального подтверждения разделений сигналов. При нажатии первой кнопки светодиод зажжется и потухнет один раз нажатии второй кнопки два раза, при нажатии девятой кнопки девять раз и все это происходит в разные интервалы времени.

Передача сигнала разной длительности представляет собой передачу сигнала с определенным временем излучения. Под этим подразумевается передача сигнала по длительности свечения. Каждой клавише соответствует программа реализующая свечение светодиода в определенном интервале времени. Самым долгим свечением определяется цифра девять.

На макете выполненного устройства установлен светодиод красного спектра для демонстрации его работы устройства и переключения режимов модуляции. Это было сделано из-за того, что ИК светодиод не виден человеческим глазом и невозможно подтвердить передачу им сигнала.

2.3 Инфракрасный фотодиод

В данной работе был использован фотодиод инфракрасного спектра номиналом BL-L512PD [12]. Рабочий диапазон длины волны данного фотодиода равен 940 нм, что соответствует используемому светодиоду ИК излучения L-34F3C. Напомним, что в инфракрасный спектр входят все длины волн, начиная с 760 нм и выше. Таким образом, инфракрасный канал связи между передатчиком и приемником разрабатываемого пульта дистанционного управления излучает ИК свечение на длине волны 940 нм. Это является одним из главных аргументов его выбора и использования в данной работе. К ним же относится низкая стоимость устройства, и простота монтажа на макетную плату, так как он занимает один порт вывода. В таблице 2 приведены технические характеристики фотодиода ИК излучения BL-L512PD [12].

Таблица 2 – Технические характеристики BL-L512PD [12]

Название характеристики	Показатель
Рассеиваемая мощность P, мВт	150
Обратное напряжение, В	35
Длина волны, нм	940
Напряжение разомкнутой цепи, В	0,4
Ток короткого замыкания, мкА	2
Обратное пробивное напряжение, В	170
Угол обзора, град	35
Способ монтажа	В отверстие
Рабочая температура, С	-40..85

На рисунке 9 изображен фотодиод ИК излучения BL-L512PD.

Рисунок 9 – Фотодиод ИК излучения BL-L512P [12]

2.4 Светодиодный индикатор

В приемном устройстве в качестве аппаратуры вывода информации был использован светодиодный индикатор фирмы Kingbright номиналом SC56-11EWA [13]. Данный индикатор используется для проведения визуального контроля на процессом передачи данных. Иными словами на него выводиться номинал кнопки нажатой на передатчике. Использование данного индикатора помогает более оперативно реагировать на какие либо изменения. Имеется десять ножек по пять с обеих сторон индикатора, по одной земле на каждой из них, так как он является устройством с общим катодом. Монтаж данного елемента довольно прост. Достаточно вставить в макетную плату и заземлить, и каждую ножку через сопротивление идущее в комплекте номиналом 110 Ом подключить на соответствующие порты [14]. Подавая логическую единицу на один из портов будет загораться один или несколько сегментов индикатора. Напоследок следует отметить что корпус данного элемента изготовлен из влагостойкой пласмассы. К тому же, SC56-11EWA защищен от любого внешнего воздействия и в часности попадания внутрь. На рисунке 10 изображен светодиодный индикатор.

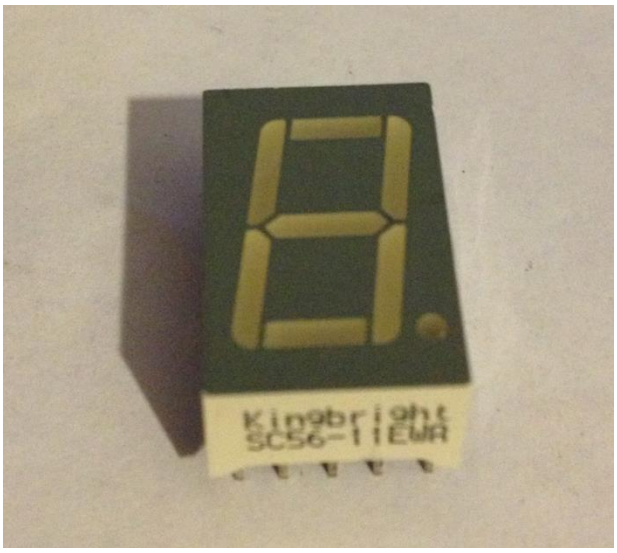

Рисунок 10 – Светодиодный индикатор SC56-11EWA [13]

В таблице 3 приведены характеристики светодиодного индикатора SC56-11EWA.

Таблица 3 – Характеристики SC56-11EWA

Название характеристики	Показатель
Светодиодный дисплей	0,56″, HE-RED, CC
Число значащих цифр	1
Размер символа	14,2 мм

Продолжение таблицы 3	
Название характеристики	Показатель
Размер символа	14,2 мм
Цвет светодиода	яркий красный
Общий контакт	катод
Типовая интенсивность	2,2 мКд
Прямой максимальный ток	20 мА
Прямое падение напряжения	2 В
Внешняя глубина	8 мм
Внешняя длина / высота	19,05 мм
Внешняя ширина	12,5 мм
Диапазон рабочих температур, С	-40..85
Цвет фона	серый
Тип символов	8
Интенсивность при максимальном токе	10 мА
Прямое максимальное напряжение	2 В
Цвет светодиода	яркий красный
Расстояние между выводами	2,54 мм
Максимальная яркость при токе If	5,6 мКд
Минимальная яркость при токе If	2,2 мКд
Яркость на сегмент при типовом токе	2,2 мКд
Число символов	1
PIV Max	5 В
Пиковый прямой ток	20 мА
Шаг рядов	15,24 мм
Типичное напряжение питания	2 В

2.5 Усилитель Дарлингтона

Усилитель – это устройство для усиления входного сигнала (например, напряжения, тока или механического перемещения, колебания звуковых частот, давления жидкости или потока света), но без изменения вида самой величины и сигнала, до уровня достаточного для срабатывания исполнительного механизма (или регистрирующих элементов), за счет энергии вспомогательного источника [15].

Усилитель, называется именно так, не по причине, что его автор Дарлингтон [16], а потому, что выходной каскад усилителя мощности построен на дарлингтоновских (составных) транзисторах. Два транзистора одинаковой структуры соединены специальным образом для высокого усиления. Такое соединение транзисторов образует составной транзистор, или транзистор Дарлингтона – по имени изобретателя этого схемного решения. Такой транзистор используется в схемах, работающих с большими токами (например, в схемах стабилизаторов напряжения, выходных каскадов усилителей мощности) и во входных каскадах усилителей, если необходимо обеспечить большой импеданс. Составной транзистор имеет три выводв (база, эмиттер и коллектор), которые эквивалентны выводам обычного одиночного транзистора. Коэффициент усиления по току типичного составного транзистора, у мощных транзисторов составляет порядка 1000 и у маломощных транзисторов порядка 50000 [16].

2.6 Цифровые микроконтроллеры

В моем ПДУ микросхемой, которая будет отвечать за управление всеми внешними устройствами является микроконтроллер.

Микроконтроллер – электронная интегральная микросхема, с цифровой логикой функционирования предназначенная для сбора, хранения и автоматической обработки информации и управления внешними процессами.

Микроконтроллер состоит из: памяти (устройство хранения цифровой информации), процессор (устройство обработки цифровой информации), порты (устройства, с помощью которых обеспечивает электрическую и логическую конкретных внутренних элементов (памяти, процессора, встроенных специальных модулей) микроконтроллера с внешней средой для обмена информацией (передачи электрических сигналов)) [17].

Рассмотрим МК фирмы «ATMEL». В середине 90-х годах, «ATMEL» начала развитие нового направления деятельности – производство высокопроизводительных 8-разрядных RISK-микроконтроллеров для встраиваемых приложений, которая объединена одним названием AVR [18]. МК AVR имеют высокий спрос, потому что привлекают разработчиков своими достоинствами, такими как цена, быстродействие, энергопотребление, удобными режимами программирования, также имеют доступ к программно-аппаратным средствам поддержки и широкую номенклатуру выпускаемых кристаллов [18]. Микроконтроллеры данной серии – удобный инструмент, который позволяет создавать современные высокопроизводительные и экономичные встраиваемые контроллеры многоцелевого значения. Широко применяются в бытовой технике, автомобильной электронике, сетевых картах и материнских платах компьютеров, в мобильных телефонах. Ниже на рисунке 11 представлен микроконтроллер фирмы ATMEL.

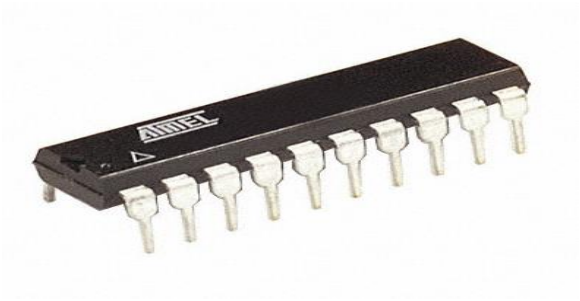

Рисунок 11 – Микроконтроллер фирмы ATMEL [19]

Серия AVR-микроконтроллеры делятся на три семейства: classic AVR – базовая линия микроконтроллеров, Mega AVR – микроконтроллеры для сложных приложений, которые требуют большого объема памяти программ и данных, Tiny AVR – микроконтроллеры с низкой стоимостью в 8–выводным исполнении [20].

Для моей работы подходит МК семейства Mega AVR, потому что имеет возможность для работы с большим объемом программ, памяти и данных. Базовая структура для микроконтроллеров семейства AVR представлена ниже на рисунке 12.

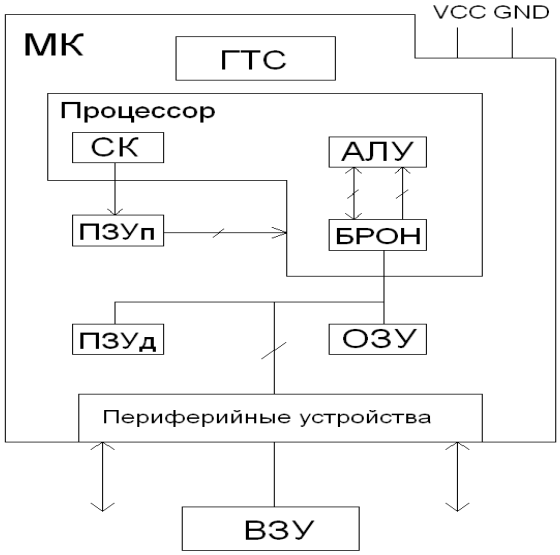

Рисунок 12 – Обобщенная структурная схема микроконтроллера

Микроконтроллер состоит: генератор тактового сигнала (ГТС), процессор, постоянное запоминающее устройство для хранения программы (ПЗУп), оперативное запоминающее устройство (ОЗУ), которое нужно для хранения данных, постоянное запоминающее устройство для хранения данных (ПЗУд), набор периферийных устройств, для ввода и вывода данных, управляющих сигналов, выполнения других функций.

Выводы VCC и GND предназначены для подключения источника напряжения питания МК. В зависимости от типа микроконтроллера может быть подключено внешнее запоминающее устройство для хранения данных (ВЗУ).

В состав процессора входит: счетчик команд (СК), арифметико-логическое устройство (АЛУ), блок регистров общего назначения (БРОН) [21].

Порты ввода/вывода микроконтроллеров семейства MEGA. С внешним миром микроконтроллер общается через порты ввода вывода. От превышенных и отрицательных значений напряжения порт защищён внутренними диодами. Эти диоды слабы и защищают порт от перепадов всего в 1-2 В, Большие перепады способны, привести к выгоранию порта [22].

На рисунке 13 показана схема микроконтроллера Atmega8 в корпусе PDIP. Порты ввода/вывода обозначены следующим образом PXY, где X - имя порта, а Y - номер бита в этом порту (например, вывод PD0 является нулевым битом порта D).

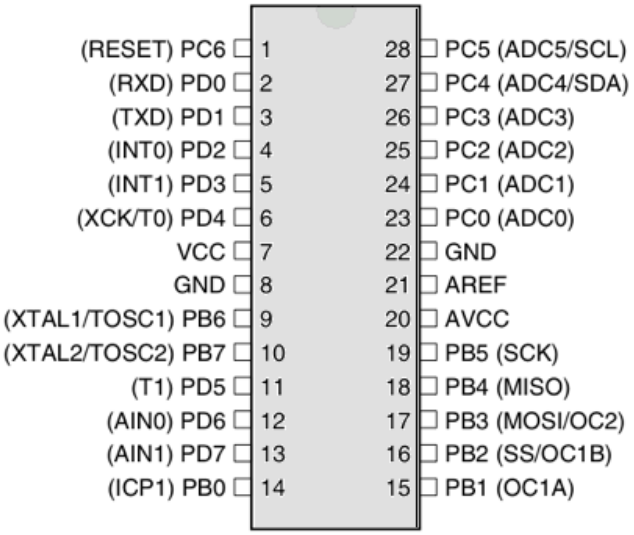

Рисунок 13 – Микроконтроллер Atmega8 [22]

В микроконтроллере Atmega8 используются 3 порта B, C, D. Порты B и D имеют 8 бит, а порт C 7 бит и являются двунаправленными. Каждый вывод порта может быть индивидуально сконфигурирован как вход или выход, причем в случае функционировании в качестве входа к нему может быть подключено подтягивающее сопротивление (рисунок 14).

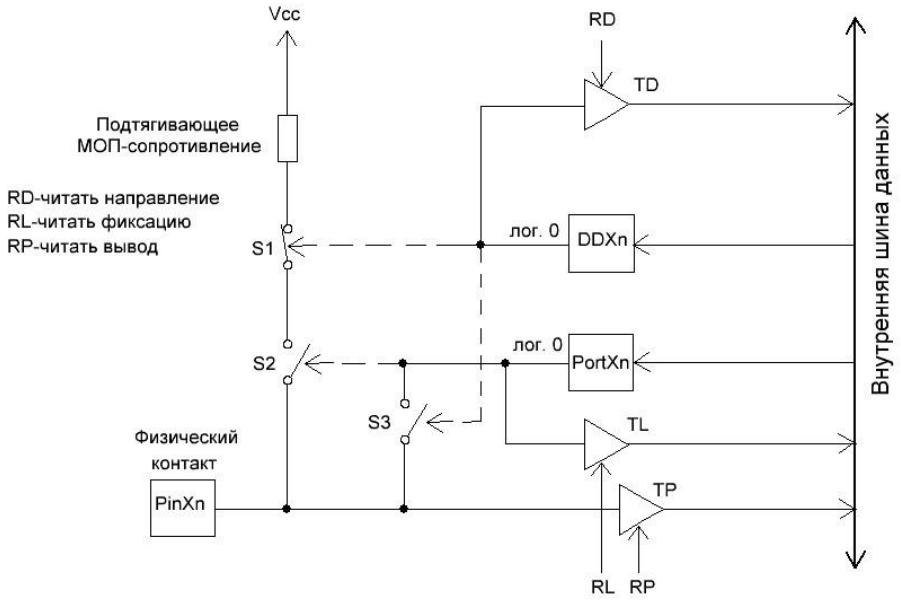

Рисунок 14 – Принципиальная схема вывода порта

На рисунке 14 показана схема, по которой построены порты микроконтроллеров семейства AVR. Так что она одинакова для Atmega8 и Atmega8535. Буква “X” в обозначении вывода регистра порта и регистра направления передачи данных используется вместо обозначения порта, то есть, вместо нее можно поставить буквы “B”, “C” или “D” и для Atmega8535 дополнительный порт “A” номера разрядов 0-7 внутри регистра представлены буквой “n”[23].

Положение ключей S1-S3, показанные на рисунке 8, соответствуют сигналам низкого уровня ны выходах регистров DDXn PortXn.

Обращение к каждому из портов осуществляется по трем различным адресам в области ввода/вывода [24]:

а) Регистр направления передачи данных (DDX) определяет назначение вывода порта: вход или выход. То есть если разряд “n” в регистре DDXn содержит сигнал низкого уровня, то соответствующий вывод сконфигурирован как вход. Ключ S3 (замыкающий контакт) в этом случае разомкнут и определяет регистр PortXn от вывода PinXn. Ключ S1(размыкающий контакт) при этом закрыт и подключает подтягивающее сопротивление к выводу PinXn, если ключ S2 замкнут посредством логической 1 ны выводе регистра PortX. Если же разряд “n” в регистре DDXn содержит сигнал высокого уровня, то соответствующий вывод сконфигурирован как выход. Ключ S3 (замыкающий контакт) в этом случае замкнут и отделяет регистр PortXn от вывода PinXn. Ключ S1 (размыкающий контакт) при этом разомкнут и отделяет подтягивающее сопротивление от вывода PinXn, независимо от положения ключа S2 и значения на выходе регистра PortXn.

б) В регистр порта PortX в том случае, если вывод выполняет роль выхода, записывается значение, подлежащее выводу. В том случае, когда вывод выполняет роль входа, ключ S2 замкнут, и подтягивающее сопротивление может быть подключено к входному контакту.

в) Логический уровень на выводе PinXn может быть считан по третьему адресу. Схема управления в этом случае вырабатывает сигнал считывания RP (Read Pin – читать вывод), который через драйвер TP подключает вывод непосредственно к внутренней шине данных.

Регистр направления передачи данных и регистр порта доступны для чтения и записи, а информация на выводах аппаратного обеспечения может быть только считана [24]. Для того чтобы выполнить операцию чтения, схема управления выдает сигналы RD (Read Rirection – читать направление), RL (Read Latch – читать фиксацию) и, соответственно, RP, которые через драйверы TD, TL и TP передают соответствующие значения на внутреннюю шину данных [24].

Выходной усилитель-формирователь порта микроконтроллера семейства AVR при низком уровне сигнала на выходе в состоянии принимать входные токи силой до 20 мА и благодаря этому, например, напрямую управлять светодиодами, подключенными к питающему напряжению [25].

В данной работе микроконтроллер Atmega8 является управляющей микросхемой приемника. Порты PD0, PD1, PD2, PD3, PD4, PD5, PD6 и PC1 настроенны на вывод, PB0, PC0 и PC2 на ввод. На порты D подается логическая единица что бы зажечь сегмент светодиодного индикатора, PC1 запрограммирован на питание тумблера- переключения типа сигнала, а PC0 и PC2 запрограммированны для проверки положения тумблера, то есть на каком порте появилась логическая единица значит микроконтролер работает в режиме модуляции для данного положения тумблера. Порт PB0 настроен на прием с выхода фотодиода на порте принимается единица, как только что то на него светит появляется 0.

В данной работе светодиод будет подключен к порту PC1, порты PD0, PD1, PD2, PD3 будут подсоединены к горизонтальным магистралям клавиатуры, а PD4, PD5, PD6 к вертикальным магистралям. К значительным плюсом данного микроконтроллера входит то что порты A, B, C, D расположенны последовательно. Если разделить микроконтроллер на 4 части то каждый занимает свою четверть. Так как мы обращаемся ко всему порту одновременно а порт включает в себя 8 ножек от PB0 до PB7, это упрощает программную реализацию. С характеристиками ATmega8535 можно ознакомиться в приложении В.

3 Схемная реализация передающей части ПДУ

На рисунке 15 изображена схема передатчика.

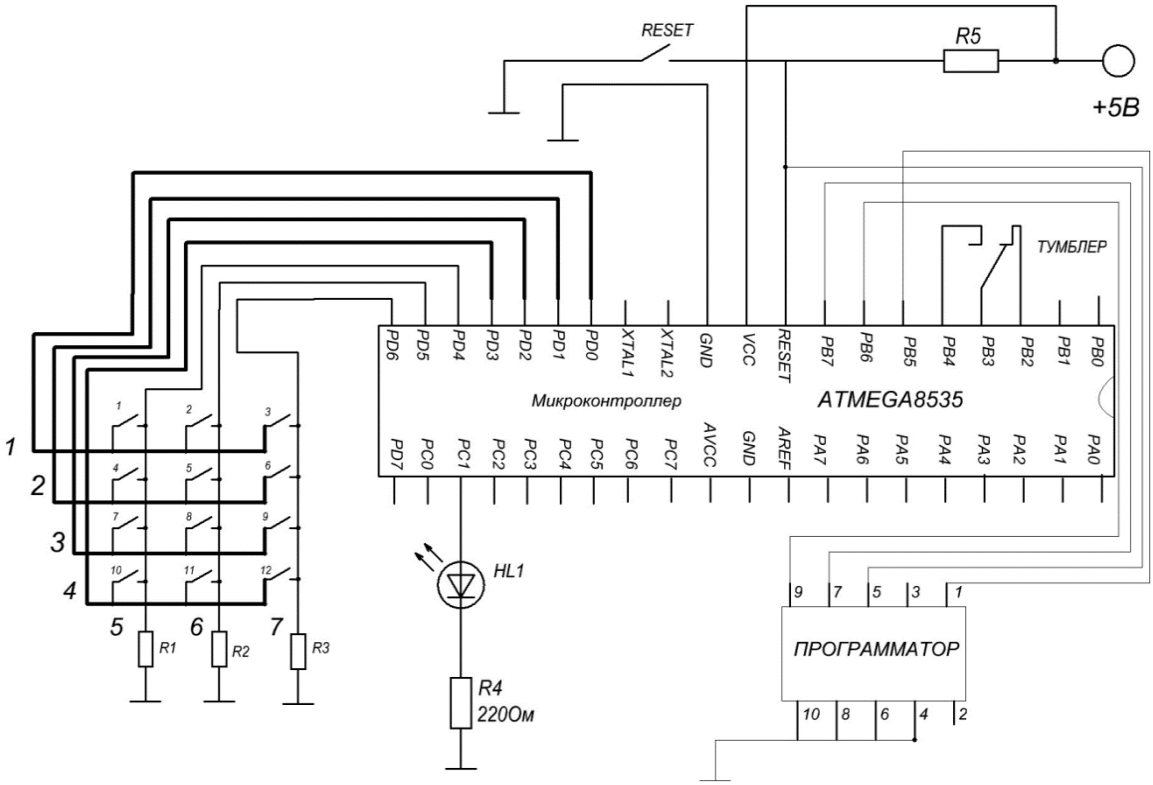

Рисунок 15 – Схема передатчика

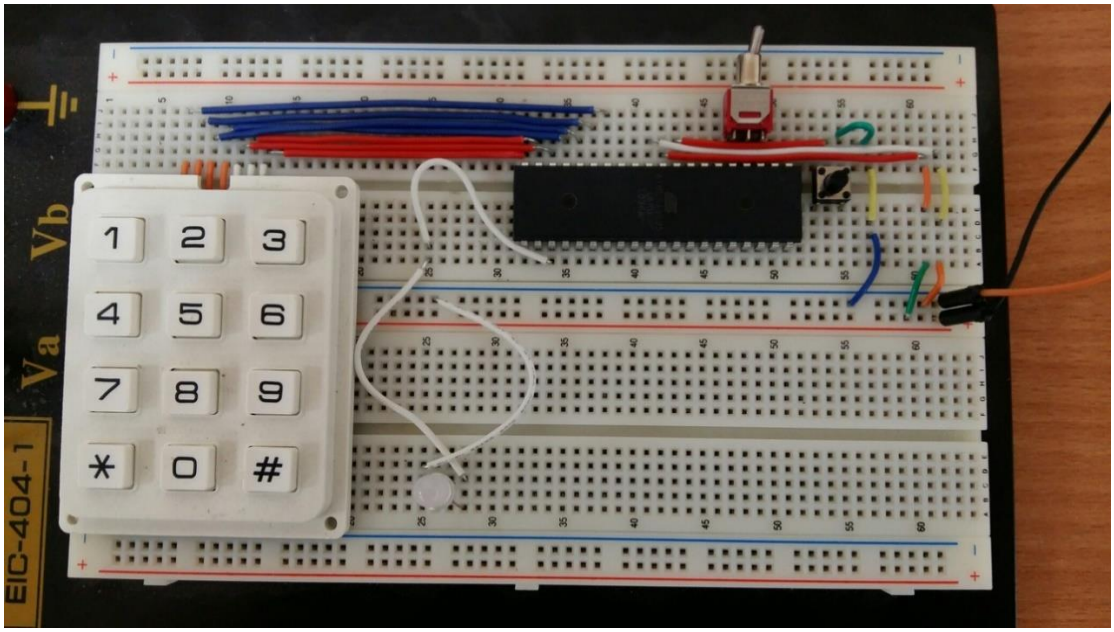

Рисунок 16 – Внешний вид макета передающей части ПДУ

На рисунках 15 и 16 представлена схема и макет передающего устройства пульта дистанционного управления.

В нашей схеме на каждый из портов вывода клавиатуры (PD0, PD1, PD2) по отдельности мы падаем логическую единицу (напряжение), и опрашиваем порты ввода (PD4, PD5, PD6) при появлении на одном из портов логической единицы мы переходим на подпрограмму определения кнопки. Определив, какая клавиша нажата программа посылает на вывод PC1 напряжение. Тем самым светодиод горит, с длительностью заданной для данной кнопки. После программа переходит на прерывание равное самой большой длительности, это делается для того чтобы при нажатии другой клавиши приемник принял правильно сигнал, а не наложение двух, что повлечет неправильно определение длительности и нажатой клавиши. Кнопка RESET подает напряжение на порт RESET, тем самым микроконтроллер перезапускает программу заново. Это в режиме, когда тумблер передатчика, стоит в режиме передачи сигнала по длительности излучения. Если же он стоит в частотном режиме, то пульт после определения кнопки начинает загораться и потухать. Номинал кнопки и есть количество мерцаний светодиода. Как видно на схеме светодиод подключен с ограничителем тока в виде резистора. Номинал резистора был вычислен по закону Ома и равен 220 Ом. Если использовать прямое подключение к источнику питания, то по причине неконтролируемого значения силы тока светодиод быстро перестанет функционировать должным образом.

4 Программное обеспечение передающей части ПДУ

На рисунке 17 рассмотрим блок схему программы передатчика.

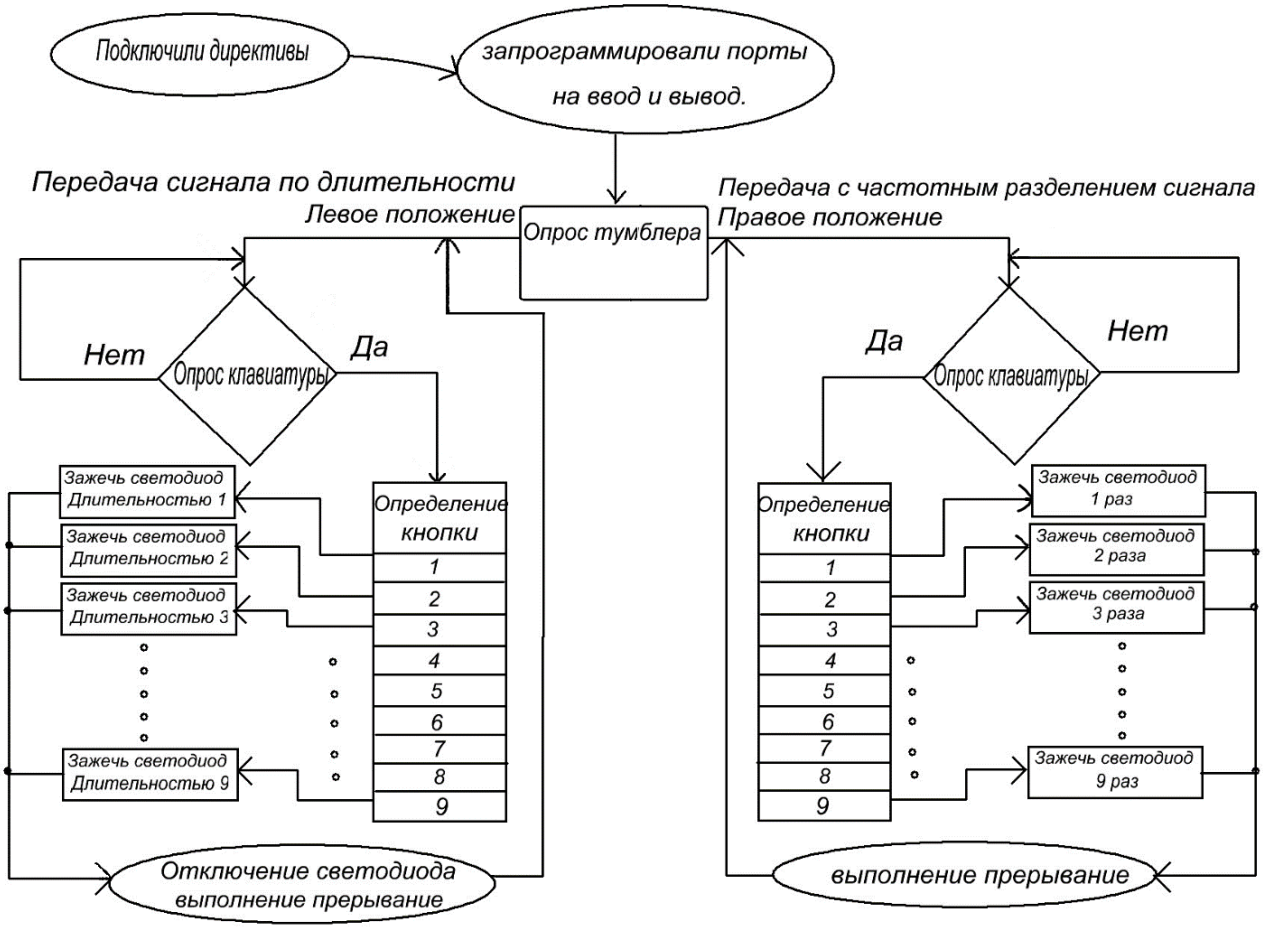

Рисунок 17 – Блок схема программы для передатчика

Данная программа состоит из:

а) директив;

б) подготовки к началу работы (программирование портов на ввод/вывод);

в) главной программы (опрос тумблера);

г) подпрограммы опроса клавиатуры;

д) подпрограмма, реализующая мигание светодиода;

е) подпрограмма задержки (прерывание).

Директива – это исполнительный код, который выполняет компилятор. Она переводит исходный текст в .hex формат. Любая программа, написанная на ASSEMBLER, будет начинаться именно с них. Поясним подробнее директивы, которые были использованы при написании программы. Например, «.include “m8def.inc» – это директива включает имена всех регистров, устройств. За ними завязаны номера всех элементов, к которым можно обратиться. Далее идет «.equ. LED=PC1» – это директива присвоила порту PC1 имя LED, так что в

дальнейшем можно будет использовать эти оба названия и процессор не будет считать это ошибкой. Директива «.cseg» – обращается к компилятору, чтобы он выбирал сегмент и место куда записывать, в память команд или память данных. «.org0» – записывает программу в память программ, начиная с нулевой ячейки памяти (0 означает ячейку памяти, с которой начинается запись).

Подготовка заключается в том, что мы переносим информацию о количестве оперативной памяти, хранящейся в регистре RAMEND в регистр SPI, который показывает количество свободных ячеек в ней. Так как оба регистра 16 битные, то нет команды для того чтобы скопировать из одного в другой. Поэтому мы взяли 8 битный регистр r16, который будет промежуточным звеном. Первая команда выделяет младший байт из константы RAMEND и загружает в регистр r16. Далее вторая команда передает содержимое из r16 в младший байт регистра указателя стека. По такому же принципу передается старший байт константы RAMEND и загружается в старший байт указателя стека. Далее в подготовке идет установка портов на ввод и на вывод информации. На примере одного порта рассмотрим этот процесс. Первым делом выбираем регистр, к примеру, r17 пользуясь командой ldi (предназначенной для передачи на внутренние устройства) загружаем туда единицу, далее уже командой out (для передачи на внешние устройства) загружаем ее в регистр направления (DDR) нужного нам порта. В данном случае порт будет работать на вывод информации. Если же в регистр направления (DDR) порта послать логический ноль, то он будет работать на ввод информации.

Главная программа. Принцип ее работы заключен в том, что наш тумблер подсоединяется тремя ножками на порты PB2, PB3, PB4. На PB3 постоянно подается логическая единица. И в какую сторону ключ замкнут, на порты PB2, PB4 запрограммированные на прием сигнал, попадает логическая единица. Тем самым мы переходим на подпрограмму, которая соответствует данному положению тумблера. Технически это выглядит так, командой Sbi – подается логическая единица на порт PB3, после, переходим на проверку регистра приема PINB соответствующего PB2. Если регистр пустой команда SBIC пропускает следующую команду и идет проверять PINB порта PB4, если же в регистре PINB порта PB2 логическую единицу, то переходит на реализацию программы по длительности свечения. Во втором положении если в PINB для порта PB4, присутствует логическая единица, то выполняется передача с частотным разделением сигналов. Тумблер может находиться только в одном положении из двух.

Подпрограммы опроса клавиатуры. Принцип ее построения заключатся в том, что у нас есть три порта, которые работают на ввод, это PD0, PD1, PD2. На нулевой порт PD0 подается логическая единица, далее идет переход на подпрограмму реализующую проверку горизонтальных магистралей PD5, PD6, PD7. После проверки процессор опять возвращает к

главной программе и продолжает ее выполнение с места, где ее прервали. Далее на нулевой порт подается ноль командой. Такой же цикл повторяется для первого и второго порта PD1, PD2. Таким образом, реализуется постоянное сканирование кнопок клавиатуры.

Подпрограмма мигания светодиода с частотным разделением в зависимости от нажатой кнопки. Всего у нас их девять прерываний по числу кнопок. При нажатии одной из них в регистр r23 записывается число равное номеру кнопки, после на порт PORTC1=LED командой SBI подается единица (светодиод горит). Далее следует переход на подпрограмму реализующую задержку примерно четверть секунды. Процессор возвращается в подпрограмму, выполняя команду CBI подавая на PORTC1 ноль (светодиод не горит). После того как светодиод отключили процессор отнимает единицу командой DEC от числа, хранящегося в r23 и сравнивает получившееся значение с нулем выполняя команду BRNE. Если значение равно нулю процессор заканчивает подпрограмму мигания светодиода и возвращается в подпрограмму проверки клавиатуры. Если же не равно нулю, BRNE отправляет процессор на метку включения светодиода. И цикл будет повторяться пока, не выполнятся все мигания, запрограммированные для нажатой кнопки.

Для горения светодиода по длительности в подпрограммы определения кнопки записывается регистр R22 с константами 3 10 14 18 и т.д. Для каждой нажатой кнопки константа разная. Чем больше номинал нажатой клавиши, тем больше константа. В подпрограмме реализующей горение светодиода на порт PC1 подается логическая единица – светодиод горит, далее подпрограмма задержки, соответствующая доле секунды, прокручивает прерывание. И длительность прерывание определяется данной константой, загруженной в R22. Программа прерывания каждый раз, когда подходит к концу цикла команда DEC отнимает от константы единицу, а команда BRNE сравнивает с нулем. При несоответствии с нулем прерывание начинается заново. При этом светодиод горит. Когда константа будет равна нулю, светодиод потухнет.

Подпрограмма прерывания производит процессором простейшие математические действия типа вычитание с определенными временными рамками. В данной работе мы использовали три регистра r24, r25, r26. Загрузили в каждый из них свою константу равную 40. Далее написали условия, при которых от регистра r26 отнимаем единицу, выполняем команду DEC и сравниваем полученное значение с нулем командой BRNE. Если не равно переходим на метку вычитания единицы, и так пока не получим ноль в r26. Далее отнимаем единицу от r25, с периодом равным полному циклу вычитания от r26. Как только r25 будет равно нулю, то от константы, находящейся в r24 отнимется единица. И так будет продолжаться, пока r24 не станет равным нулю. Смысл в том, что один период полного цикла для одной константы, является временем между вычитаниями для другой.

5 Схемная реализация приемной части ПДУ

На рисунках 18 и 19 изображена схема и макета приемной части пульта.

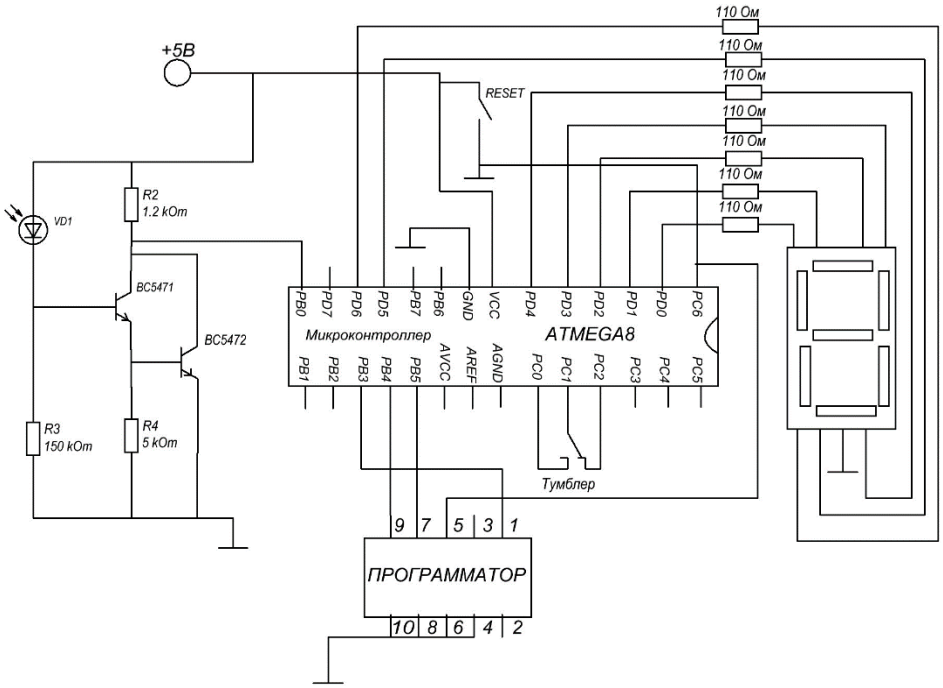

Рисунок 18 – Схема приемника

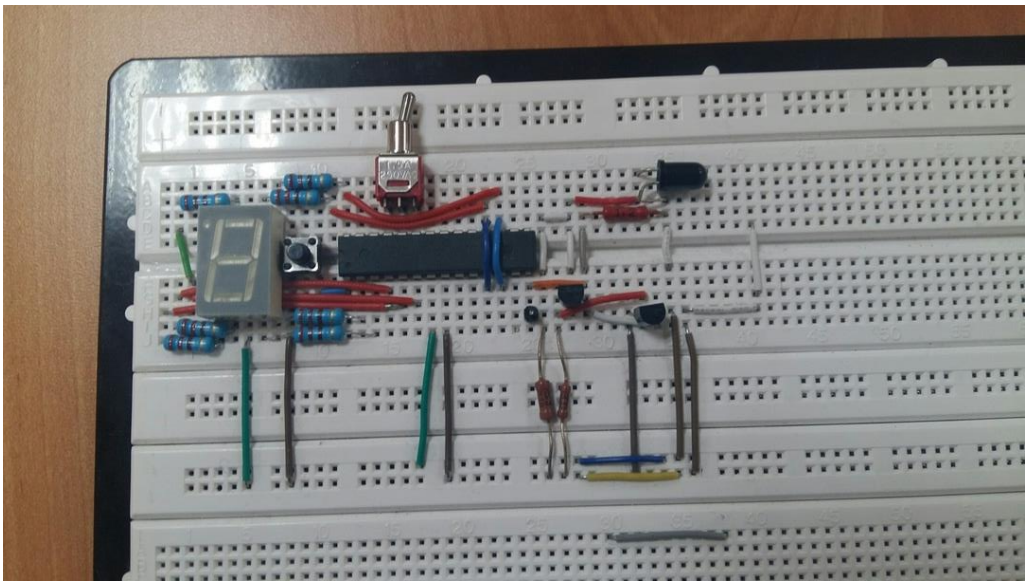

Рисунок 19 – Макет приемной части ПДУ

Начнем с того что в микроконтроллере порт D настроен на вывод напряжения, а PB0 на ввод. Тем самым PB0 работает на прием, а PD0, PD1, PD2, PD3, PD4, PD5, PD6 на передачу. Когда на фотодиод ничего не светит, то порт PB0 принимает напряжение примерно равное 5 вольт или логическую единицу. При появлении инфракрасного излучения, фотодиод увеличивает свое сопротивление, и порт получает логический ноль с выхода усилителя.

Включается программа отчета длительности свечения. И когда свечение пропадает, программа определяет, в какой диапазон оно попало. Каждому диапазону соответствует своя цифра от 1-9. В зависимости от цифры на определенные порты D подается напряжение и высвечивается цифра, нажатая на передатчике. В режиме частотной передачи сигнала, отслеживается количество падений напряжений на порте PB0, далее
зажигается светодиодный индикатор с цифрой, которая ему соответствует.

6 Программное обеспечение приемной части ПДУ

На рисунке 20 изображена блок схема программы.

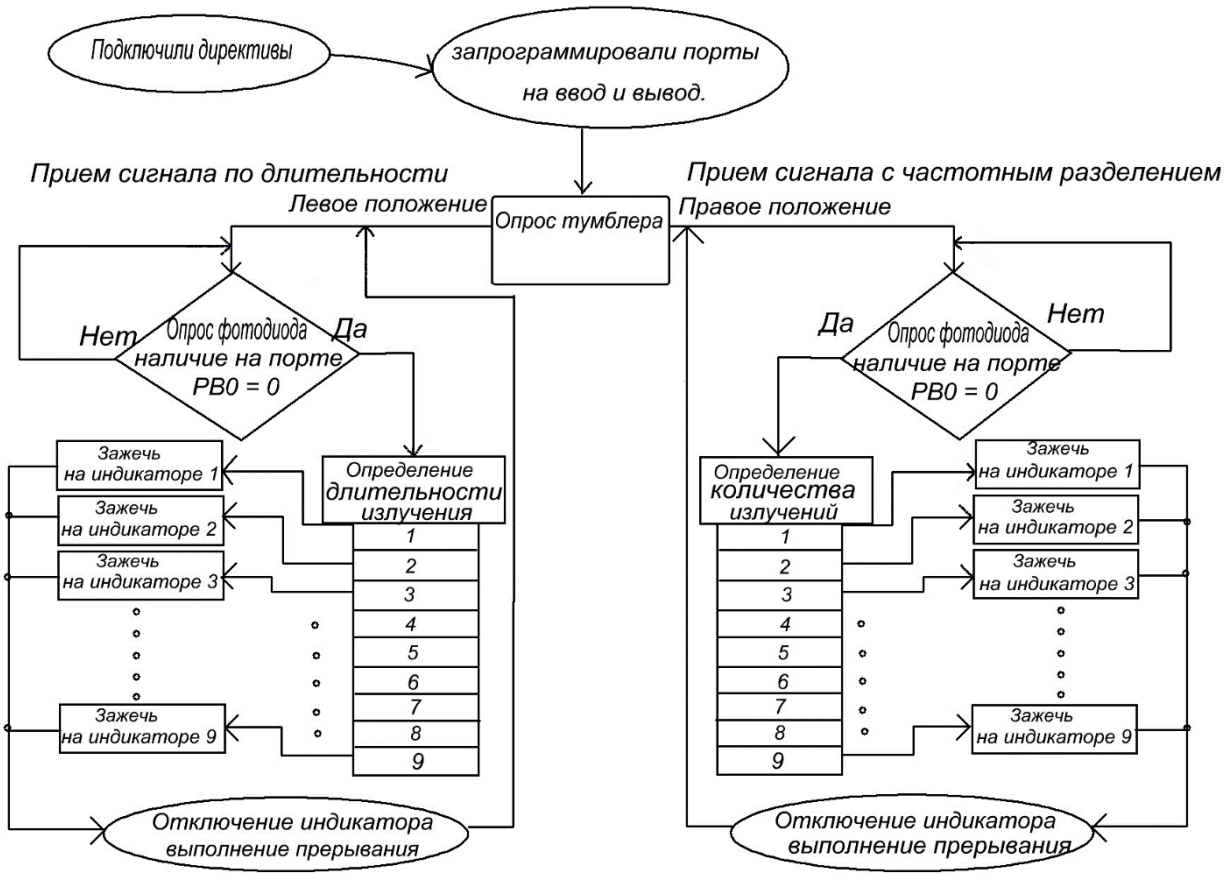

Рисунок 20 – Блок схема программы приемника

Данная программа состоит из: директив, подготовки к началу работы (программирование портов на ввод/вывод), главной программы (опрос тумблера), определение диапазона принимаемых сигналов, подпрограмма, реализующая высвечивание цифры на индикаторе, подпрограмма задержки (прерывание).

Список директив остался прежним так что начнем с подготовки к началу работы. Настроили порты PB0, PC2,PC0 на ввод информации в микроконтроллер. Все порты от PD0 до PD6 и PC1 запрограммировалли вывод. Опрос тумблера осуществляется таким же образом как и в передатчике.

Рассмотрим сам принцип приема. С выхода усилителя дарлингтона подается логическая единица. И порт PB0 ее принимает, при этом в регистре PINB высвечивается ее наличие. При попадании на фотодиод инфракрассного излучения, он меняет свое сопротивление и на выходе усилителя будет 0. Он же появится и в регистре PINB тогда программа переходит на определение диапазона передаваемого сигнала. Если устройство работает на передачу сигналов разной длительности то программа отчитывает время

нахождения логического нуля в регистре PINB. Это выглядит ступенчатым образом. Если длительность сигнала не входит в первый диапазон, переводит его во второй, далее третий и так пока сигнал не прекратится в каком то из девяти диапазонов. На каком диапазоне сигнал прекратился. Цифра соответствующая ему будет выведена на индикатор, с помощью подачи логической единицы в разные выводы порта PB. После идет программа задержки во время которой светится индикатор послее ее окончания он гаснет, и программа снова переходит в режим опроса PINB. В режиме с чистотным разделением сигналов, после приема на порте PB0 логического нуля. Программа переходит первую подпрограмму, всего их девять. Они состоят из прерывания и опроса порта PB0 а также из двух переходов, на следующию подпрограмму и индикация цифры. Если при выполнении одной из них логическая единица не сменилась логическим нулем то она переходит на программу индикации цифры. Если же сменилась то на последующую подпрограмму. При индикации подаются логические единицы на разные вывроды порта PD0 тем самым загорается цифра соответствующая количеству принятых сигналов. Пока идет индикация выполняется задержка чтобы могли сравнить высвеченную цифру и номер кнопки. После ее окончания выводы PD0 обнуляются и программа переходит на опрос порта PB0.

ЗАКЛЮЧЕНИЕ

В ходе выполненной работы были изучены:

а) архитектура микроконтроллера на системе RISC;

б) принципы программирования;

в) программные средства: AVRstudio4;

г) разработана схема, программа и макет ПДУ.

Вывод: В данной работе была изученна структура микроконтроллеров, возможности их использования, принцип программирования. А так же был разработан макет пульта дистанционного управления. Написана программа на языке ASSEMBLER для микроконтроллера передатчика и приемника, тем самым были полученны практические знания управления процессами в микроконтроллере. В заключении можно отметить, что реализация данного устройства может применятся для управления внешними процессами.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Цифровое устройство. URL: https://dic.academic.ru/dic.nsf/ruwiki/1889373 (Дата обращения 22.11.2018).

2 Цифровой сигнал. URL: https://ru.wikipedia.org/wiki/Цифровой_сигнал (Дата обращения 22.11.2018).

3 Что такое микроконтроллеры – назначение, устройство, софт. URL: http://electrik.info/main/automation/549-chto-takoe-mikrokontrollery-naznachenie-ustroystvo-princip-raboty-soft.html (Дата обращения 23.11.2018).

4 Микроконтроллеры семейства AVR. URL: http://biosgid.ru/osnovy-ustrojstva-pk/mikrokontroller-avr.html (Дата обращения 14.12.2018).

5 Пульт дистанционного управления. URL: https://ru.wikipedia.org/wiki/Пульт_дистанционного_управления (Дата обращения 23.11.2018).

6 Классификация проводных каналов связи. URL: https://www.sviaz-expo.ru/ru/articles/2016/provodnye-kanaly-svyazi/ (Дата обращения 24.11.2018).

7 Радиоканал. URL: https://dic.academic.ru/dic.nsf/enc_tech/990/радиоканал (Дата обращения 24.11.2018).

8 Дистанционное управление URL: https://ru.wikipedia.org/wiki/Дистанционное_управление (Дата обращения 22.11.2018).

9 Инфракрасный канал связи URL: https://life-prog.ru/view_zam2.php?cat=2&id=40&page=6 (Дата обращения 23.11.2018).

10 Пульт дистанционного управления. URL: https://dic.academic.ru/dic.nsf/ruwiki/249302 (Дата обращения 25.11.2018).

11 Подключение к микроконтроллеру клавиатуры матричного типа URL: http://www.labfor.ru/guidance/mpu-leso1/3 (Дата обращения 14.12.2018).

12 Фотодиод ИК BL-512PD URL: https://www.chipdip.ru/product/bl-l512pd (Дата обращения 03.12.2018).

13 Семисегментный индикатор SC56-11EWA URL: https://www.chipdip.ru/product/sc56-11ewa (Дата обращения 14.12.2018).

14 SC56-11EWA –7-сегментный светодиодный дисплей URL: https://ru.farnell.com/kingbright/sc56-11ewa/display-seven-segment-19-05mm/dp/1142441 (Дата обращения 14.12.2018).

15 Усилитель URL: https://ru.wikipedia.org/wiki/Усилитель (Дата обращения 13.12.2018).

16 Усилитель Дарлингтона URL: http://amplif.ru/publ/pro_usiliteli/usilitel_darlingtona/4-1-0-154 (Дата обращения 14.12.2018).

17 Устройство микроконтроллера AVR. URL: https://myrobot.ru/stepbystep/mc_architecture.php (Дата обращения 14.12.2018).

18 Микроконтроллеры фирмы Atlem. URL: http://www.gaw.ru/html.cgi/txt/ic/Atmel/micros/index.htm (Дата обращения 14.12.2018).

19 GETCHIPS электронные компоненты. URL: http://www.chipfind.ru/catalog/ic/microcontrollers/attiny26120pu.htm (Дата обращения 14.12.2018).

20 Виды и устройство микроконтроллеров. AVR URL: http://electrik.info/main/praktika/1333-ustroystvo-mikrokontrollerov-avr.html (Дата обращения 14.12.2018).

21 Процессор и его функции. URL: http://www.gm4.ru/pril/shamardin2/cpupodr.html (Дата обращения 14.12.2018).

22 Нумерация ножек микросхем URL: http://nintones.appspot.com/numeraciya-nozhek-mikroshem.html (Дата обращения 03.12.2018).

23 Устройство и работа портов ввода-вывода микроконтроллеров AVR часть 1. URL: https://habr.com/post/253213/ (Дата обращения 14.12.2018).

24 AVR. Учебный курс. Устройство и работа портов ввода-вывода. URL: http://easyelectronics.ru/avr-uchebnyj-kurs-ustrojstvo-i-rabota-portov-vvoda-vyvoda.html (Дата обращения 14.12.2018).

25 AVR для начинающих. Урок 3. Порты ввода-вывода. URL: http://mkprog.ru/avr/avr-dlya-nachinayushhih-urok-3-porty-vvoda-vyvoda.html (Дата обращения 14.12.2018).

ПРИЛОЖЕНИЕ А (СПРАВОЧНОЕ)

Текст программы для передающего устройства.

```
.include "m8def.inc"                ;присоединение кода из файла (деректива)
.EQU LED = PC1
.cseg                               ;выбор сегмента для записи кода (деректива)
.org 0                              ;установка начального адреса программы (деректива)
rjmp RESET          ;Программный переход относительно текущего адреса на метку RESET
nop ;rjmp EXT_INT0                  ;INT0 External Interrupt Request 0
nop ;rjmp EXT_INT1;                 ;INT1 External Interrupt Request 1
nop ;rjmp TIMER2_COMP               ;TIMER2 COMP Timer/Counter2 Compare Match
nop ;rjmp TIMER2_OVF                ;TIMER2 OVF Timer/Counter2 Overflow
nop ;rjmp TIMER1_CAPT               ;TIMER1 CAPT Timer/Counter1 Capture Event
nop ;rjmp TIMER1_COMPA                      ;TIMER1 COMPA Timer/Counter1 Compare Match A
nop ;rjmp TIMER1_COMPB              ;TIMER1 COMPB Timer/Counter1 Compare Match B
nop ;rjmp TIMER1_OVF                ;TIMER1 OVF Timer/Counter1 Overflow
nop ;rjmp TIMER0_OVF                ;TIMER0 OVF Timer/Counter0 Overflow
nop ;rjmp SPI_STC           ;SPI, STC Serial Transfer Complete
nop ;rjmp USART_RXC                 ;USART, RXC USART, Rx Complete
nop ;rjmp USART_UDRE                ;USART, UDRE USART Data Register Empty
nop ;rjmp USART_TXC                 ;USART, TXC USART, Tx Complete
nop ;rjmp ADC_CC            ;ADC ADC Conversion Complete
nop ;rjmp EEPROM_RDY                ;EE_RDY EEPROM Ready
nop ;rjmp ANA_COMP                  ;ANA_COMP Analog Comparator
nop ;rjmp TWI                       ;TWI Two-wire Serial Interface
nop ;rjmp SPM_RDY           ;SPM_RDY Store Program Memory Ready
RESET:
ldi   R16,high(RAMEND) ; В указатель стека загрузить
out   SPH,R16                ;адрес  последней  ячейки
ldi   R16,low(RAMEND) ; оперативной памяти
out   SPL,R16                ; микроконтроллера
ldi   R16,0    ;загрузили в 16 регистр 0
out   DDRD,R16  ;Скопировали содержимое регистра флагов порта D, в 16 регистр
ldi   R16,(1 << PD0)|(1 << PD1)|(1 << PD2)|(1 << PD3) ;Запрограммировали порты на вывод
out   PORTD,R16 ;Скопировали содержимое порта D, в 16 регистр
ldi   R16,0x02  ;загрузили в 16 регистр 00000010.
```

```
out   DDRC,R16 ;Сохранили содержимое регистра флагов порта C, в 16 регистр
ldi   R23,0x02 ;загрузили в 23 регистр 00000010.
rcall Reid
Reid:                    ;опрос тумблера
sbi Portb, $3     ;Загрузить единицу в порт PB3
rcall vn          ;перейти
vn: sbis pinb, $2 ;если единица перейди пропусти команду
rcall Main        ;перейти
rcall vb          ;перейти
vb: sbis pinb, $4  ;если единица перейди пропусти команду
rcall Maini       ;перейти
rjmp REID
Main:
clr   R19     ;обнуляем регистр
ldi   R18,0x10 ;загрузили в R18 00010000 (цифра 4)
Main1:
inc   R19          ;делаем счетчик столбцов
out   DDRD,R18      ;Копируем содержимое регистра флагов порта D
nop                ;пропускаем
nop                ;пропускаем
in    R17,PIND    ;копируем содержимое R17 в PIND
com   R17         ;инвертируем все биты
andi  R17,0x0F    ;выполняем операцию <логическое И>
cpi   R17,0x01   ;сравниваем содержимое регистра и константы
ldi   R20,0       ;Загрузили 0 в R20
breq  main2       ;переходит если установлен флаг нуля
cpi   R17,0x02    ;сравниваем содержимое регистра и константы
ldi   R20,1       ;Загрузили 1 в R20
breq  main2       ;переходит если установлен флаг нуля
cpi   R17,0x04    ;сравниваем содержимое регистра и константы
ldi   R20,2       ;Загрузили 2 в R20
breq  main2       ;переходит если установлен флаг нуля
cpi   R17,0x08    ;сравниваем содержимое регистра и константы
ldi   R20,3       ;Загрузили 2 в R20
breq  main2       ;переходит если установлен флаг нуля
```

```
lsl   R18         ;выполняем логический сдвиг влево
cpi   R19,3       ;сравниваем содержимое регистра и константы
brne  Main1       ;переходит если установлен флаг нуля
rjmp main         ;переходит в программу main
Main2:
Mov  R21,R20  ; копирование регистра R20 в R21
lsl  R20      ; умножение регистра на 2
add  R20,R21     ;складываем содержимое регистров и сохраняем в R20
add  R20,R19  ; кнопка=3*Nстроки+Nстолбца
rjmp Main3
Main3:            ; Проверка нажатия 1 клавиши
ldi r22, 1        ;  загрузили в регистр 2
cpi R20, 1        ;если 1 кнопка не нажата пропускаем следующую команду
breq main12         ;перейди на подпрограмму горения светодиода
rjmp main4          ;перейди на подпрограмму проверки 2 кнопки
Main4:              ;Проверка нажатия 2 клавиши
ldi r22, 2        ;загрузили в регистр 3
cpi R20, 2      ;если 2 кнопка не нажата пропускаем следующую команду
breq main12      ;перейди на подпрограмму горения светодиода
rjmp main5       ;перейди на подпрограмму проверки 3 кнопки
Main5:            ;  Проверка нажатия 3 клавиши
ldi r22, 3       ; загрузили в регистр 3
cpi R20, 3      ;если 3 кнопка не нажата пропускаем следующую команду
breq main12      ;перейди на подпрограмму горения светодиода
rjmp main6       ;перейди на подпрограмму проверки 4 кнопки
Main6:              ;Проверка нажатия 4 клавиши
ldi r22, 4       ;загрузили в регистр 8
cpi R20, 4        ;если 4 кнопка не нажата пропускаем следующую команду
breq main12       ;перейди на подпрограмму горения светодиода
rjmp main7        ;перейди на подпрограмму проверки 5 кнопки
Main7:      ;         Проверка нажатия 5 клавиши
ldi r22, 5        ;   загрузили в регистр 10
cpi R20, 5        ;если 5 кнопка не нажата пропускаем следующую команду
breq main12      ;перейди на подпрограмму горения светодиода
rjmp main8       ;перейди на подпрограмму проверки 6 кнопки
```

```
Main8:              ;  Проверка нажатия 6 клавиши
ldi r22, 6        ;    загрузили в регистр 12
cpi R20, 6          ;если 6 кнопка не нажата пропускаем следующую команду
breq main12       ;перейди на подпрограмму горения светодиода
rjmp main9        ;перейди на подпрограмму проверки 7 кнопки
Main9:                ;Проверка нажатия 7 клавиши
ldi r22, 7          ;загрузили в регистр 15
cpi R20, 7            ;если 7 кнопка не нажата пропускаем следующую команду
breq main12       ;перейди на подпрограмму горения светодиода
rjmp main10         ;перейди на подпрограмму проверки 8 кнопки
Main10:               ;Проверка нажатия 8 клавиши
ldi r22, 8        ;загрузили в регистр 17
cpi R20, 8          ;если 8 кнопка не нажата пропускаем следующую команду
breq main12        ;перейди на подпрограмму горения светодиода
rjmp main11        ;перейди на подпрограмму проверки 9 кнопки
Main11:
ldi r22, 9
cpi R20, 9        ;если 9 кнопка не нажата пропускаем следующую команду
breq main12 ;перейди на подпрограмму горения светодиода  для 9 кнопки
rjmp main
main12:                    ; начало главной программы
sbi portc, led; установка "1" на линии LED
rcall DELAY1
rjmp main                          ; переход на начало
DELAY1: ; Подпрограмма задержки.
ldi r23, 40                        ; Загрузить в
step1:  ldi r24, 40                        ; регистры
step2:  ldi r25, 10                        ; константы.
step3:  dec   r25                          ; Уменьшить на 1 содержимое регистра и
brne step3                         ; если в нем не 0 перейти на метку.
dec   r24
brne step2
dec   r23
brne step1
cbi portc, led             ; установка "0" на линии LED
```

```
rcall DELAY2
DELAY2:
ldi r23, 60                    ; Загрузить в
step4:  ldi r24, 60                ; регистры
step5:  ldi r25, 10                ; константы.
step6:  dec   r25                  ; Уменьшить на 1 содержимое регистра и
brne step6                     ; если в нем не 0 перейти на метку.
dec   r24
brne step5
dec   r23
brne step4
dec   r22
brne main12
rjmp main
Maini:
clr   R19     ;обнуляем регистр
ldi   R18,0x10 ;загрузили в R18 00010000 (цифра 4)
Maini1:
inc   R19          ;делаем счетчик столбцов
out   DDRD,R18      ;Копируем содержимое регистра флагов порта D
nop                ;пропускаем
nop                 ;пропускаем
in    R17,PIND    ;копируем содержимое R17 в PIND
com   R17         ;инвертируем все биты
andi  R17,0x0F    ;выполняем операцию <логическое И>
cpi   R17,0x01   ;сравниваем содержимое регистра и константы
ldi   R20,0       ;Загрузили 0 в R20
breq  maini2       ;переходит если установлен флаг нуля
cpi   R17,0x02    ;сравниваем содержимое регистра и константы
ldi   R20,1       ;Загрузили 1 в R20
breq  maini2       ;переходит если установлен флаг нуля
cpi   R17,0x04    ;сравниваем содержимое регистра и константы
ldi   R20,2       ;Загрузили 2 в R20
breq  maini2       ;переходит если установлен флаг нуля
cpi   R17,0x08    ;сравниваем содержимое регистра и константы
```

```
ldi   R20,3       ;Загрузили 2 в R20
breq  maini2      ;переходит если установлен флаг нуля
lsl   R18         ;выполняем логический сдвиг влево
cpi   R19,3       ;сравниваем содержимое регистра и константы
brne  Maini1      ;переходит если установлен флаг нуля
rjmp maini        ;переходит в программу main
Maini2:
Mov  R21,R20  ; копирование регистра R20 в R21
lsl  R20      ; умножение регистра на 2
add  R20,R21     ;складываем содержимое регистров и сохраняем в R20
add  R20,R19  ; кнопка=3*Nстроки+Nстолбца
rjmp Maini3
Maini3:           ;Проверка нажатия 1 клавиши
ldi r22, 2        ; загрузили в регистр 2
cpi R20, 1        ;если 1 кнопка не нажата пропускаем следующую команду
breq maini12    ;перейди на подпрограмму горения светодиода
rjmp maini4     ;перейди на подпрограмму проверки 2 кнопки
Maini4:           ;Проверка нажатия 2 клавиши
ldi r22, 4        ; загрузили в регистр 3
cpi R20, 2      ;если 2 кнопка не нажата пропускаем следующую команду
breq maini12   ;перейди на подпрограмму горения светодиода
rjmp maini5     ;перейди на подпрограмму проверки 3 кнопки
Maini5:           ;Проверка нажатия 3 клавиши
ldi r22, 6        ;загрузили в регистр 3
cpi R20, 3      ;если 3 кнопка не нажата пропускаем следующую команду
breq maini12   ;перейди на подпрограмму горения светодиода
rjmp maini6    ;перейди на подпрограмму проверки 4 кнопки
Maini6:           ;Проверка нажатия 4 клавиши
ldi r22, 8        ;загрузили в регистр 8
cpi R20, 4      ;если 4 кнопка не нажата пропускаем следующую команду
breq maini12    ;перейди на подпрограмму горения светодиода
rjmp maini7     ;перейди на подпрограмму проверки 5 кнопки
Maini7:           ;Проверка нажатия 5 клавиши
ldi r22, 10       ; загрузили в регистр 10
cpi R20, 5        ;если 5 кнопка не нажата пропускаем следующую команду
```

```
breq maini12     ;перейди на подпрограмму горения светодиода
rjmp maini8      ;перейди на подпрограмму проверки 6 кнопки
Maini8:                ; Проверка нажатия 6 клавиши
ldi r22, 12         ;   загрузили в регистр 12
cpi R20, 6        ; если 6 кнопка не нажата пропускаем следующую команду
breq maini12      ;перейди на подпрограмму горения светодиода
rjmp maini9       ; перейди на подпрограмму проверки 7 кнопки
Maini9: ;Проверка нажатия 7 клавиши
ldi r22, 14 ;загрузили в регистр 15
cpi R20, 7 ;если 7 кнопка не нажата пропускаем следующую команду
breq maini12 ;перейди на подпрограмму горения светодиода
rjmp maini10 ;перейди на подпрограмму проверки 8 кнопки
Maini10: ;Проверка нажатия 8 клавиши
ldi r22, 16 ;загрузили в регистр 17
cpi R20, 8 ;если 8 кнопка не нажата пропускаем следующую команду
breq maini12 ;перейди на подпрограмму горения светодиода
rjmp maini11 ;перейди на подпрограмму проверки 9 кнопки
Maini11:
ldi r22, 19
cpi R20, 9 ;если 9 кнопка не нажата пропускаем следующую команду
breq maini12 ;перейди на подпрограмму горения светодиода  для 9 кнопки
rjmp maini
maini12:                    ; начало главной программы
sbi   PORTC, LED                ; установка "1" на линии LED
rcall DELAY3                            ; задержка
rjmp MAINi                      ; переход на начало
DELAY3: ; Подпрограмма задержки.
ldi r23, 40                     ; Загрузить в
step7:  ldi r24, 40                     ; регистры
step8:  ldi r25, 10                     ; константы.
step9:  dec   r25                       ; Уменьшить на 1 содержимое регистра и
brne step9                      ; если в нем не 0 перейти на метку.
dec   r24
brne step8
dec   r23
```

```
brne step7
dec   r22
brne DELAY3
cbi   PORTC, LED              ; установка "0" на линии LED
rcall DELAY4
DELAY4:     ldi r23, 100                    ; Загрузить в
step10:ldi r24, 100                   ; регистры
step11:ldi r25, 10                    ; константы.
step12:dec   r25                      ; Уменьшить на 1 содержимое регистра и
brne step12                   ; если в нем не 0 перейти на метку.
dec   r24
brne step11
dec   r23
brne step10
rjmp Maini
```

ПРИЛОЖЕНИЕ Б (СПРАВОЧНОЕ)

Текст программы для приемной части ПДУ.

```
.include "m8def.inc"            ;присоидинение кода из файла (деректива)
.cseg                           ;выбор сегмента для записи кода (деректива)
.org 0                          ;установка начального адреса программы (деректива)
rjmp RESET                      ;Програмный переход относительно текущего адреса на метку
RESET
nop ;rjmp EXT_INT0              ;INT0 External Interrupt Request 0
nop ;rjmp EXT_INT1;             ;INT1 External Interrupt Request 1
nop ;rjmp TIMER2_COMP           ;TIMER2 COMP Timer/Counter2 Compare Match
nop ;rjmp TIMER2_OVF            ;TIMER2 OVF Timer/Counter2 Overflow
nop ;rjmp TIMER1_CAPT           ;TIMER1 CAPT Timer/Counter1 Capture Event
nop ;rjmp TIMER1_COMPA          ;TIMER1 COMPA Timer/Counter1 Compare Match A
nop ;rjmp TIMER1_COMPB          ;TIMER1 COMPB Timer/Counter1 Compare Match B
nop ;rjmp TIMER1_OVF            ;TIMER1 OVF Timer/Counter1 Overflow
nop ;rjmp TIMER0_OVF            ;TIMER0 OVF Timer/Counter0 Overflow
nop ;rjmp SPI_STC               ;SPI, STC Serial Transfer Complete
nop ;rjmp USART_RXC             ;USART, RXC USART, Rx Complete
nop ;rjmp USART_UDRE            ;USART, UDRE USART Data Register Empty
nop ;rjmp USART_TXC             ;USART, TXC USART, Tx Complete
nop ;rjmp ADC_CC                ;ADC ADC Conversion Complete
nop ;rjmp EEPROM_RDY            ;EE_RDY EEPROM Ready
nop ;rjmp ANA_COMP              ;ANA_COMP Analog Comparator
nop ;rjmp TWI                   ;TWI Two-wire Serial Interface
nop ;rjmp SPM_RDY               ;SPM_RDY Store Program Memory Ready
RESET:
ldi   R16,high(RAMEND) ;инициализация стека
out   SPH,R16
ldi   R16,low(RAMEND)
out   SPL,R16
ldi r18, (0<<DDB0);Сдвиг 0 на DDD5-позиций влево. DDD5=1 ("m8def.inc"). Затем загрузка.
out DDRB, r18     ;Передача содержимого регистра в порт (в регистр направления передачи
данных DDRD)
ldi r17, (1<<DDD0)|(1<<DDD1)|(1<<DDD2)|(1<<DDD3)|(1<<DDD4)|(1<<DDD5)|(1<<DDD6)
out DDRD, r17
```

```
main:
sbi Portc, $1      ;Загрузить единицу в порт PC1
rcall vn           ;перейти
vn: sbis pinc, $0   ;если единица перейди пропусти команду
rcall REID          ;перейти
rcall vb            ;перейти
vb: sbis pinc, $2   ;если единица перейди пропусти команду
rcall Maini         ;перейти
rjmp vn             ;перейти
REID:
sbis PINB, $0 ;Подпрограмма пропускания если бит в регистре PINB0=0 не установлен.
rcall Proverka ;Переход на подпрограмму реализующую задержку и горения светодиода при нажатой 1 кнопке (т.е PIND5=0)
RJMP Reid       ;Возврат в начало.
Proverka:
rcall PRER          ;Перейди на подпрограмму прерывания
sbis PINB, 0$       ;пропусти следующую команду если бит установлен
rcall Proverka1     ;Перейди на подпрограмму
rcall Main0       ;перейди на подпрограмму зажигания 1
proverka1:
rcall PRER          ;Перейди на подпрограмму прерывания
sbis PINB, 0$       ;пропусти следующую команду если бит установлен
rjmp Proverka2      ;Перейди на подпрограмму
rjmp Main1        ;перейди на подпрограмму зажигания 2
proverka2:
rcall PRER          ;Перейди на подпрограмму прерывания
sbis PINB, 0$       ;пропусти следующую команду если бит установлен
rcall Proverka3     ;Перейди на подпрограмму
rcall Main2       ;перейди на подпрограмму зажигания 3
proverka3:
rcall PRER          ;Перейди на подпрограмму прерывания
sbis PINB, 0$       ;пропусти следующую команду если бит установлен
rcall Proverka4     ;Перейди на подпрограмму
rcall Main4       ;перейди на подпрограмму зажигания 4
proverka4:
```

```
rcall PRER            ;Перейди на подпрограмму прерывания
sbis PINB, 0$         ;пропусти следующую команду если бит установлен
rcall Proverka5       ;Перейди на подпрограмму
rcall Main5       ;перейди на подпрограмму зажигания 5
proverka5:
rcall PRER            ;Перейди на подпрограмму прерывания
sbis PINB, 0$         ;пропусти следующую команду если бит установлен
rcall Proverka6       ;Перейди на подпрограмму
rcall Main6       ;перейди на подпрограмму зажигания 6
proverka6:
rcall PRER            ;Перейди на подпрограмму прерывания
sbis PINB, 0$         ;пропусти следующую команду если бит установлен
rcall Proverka7       ;Перейди на подпрограмму
rcall Main7       ;перейди на подпрограмму зажигания 7
proverka7:
rcall PRER            ;Перейди на подпрограмму прерывания
sbis PINB, 0$         ;пропусти следующую команду если бит установлен
rcall Proverka8       ;Перейди на подпрограмму
rcall Main8       ;перейди на подпрограмму зажигания 8
proverka8:
rcall PRER            ;Перейди на подпрограмму прерывания
sbis PINB, 0$         ;пропусти следующую команду если бит установлен
rcall Main9       ;перейди на подпрограмму зажигания 9
Main0:
ldi  R28,0b00011000   ;загрузили 0b00011000 в R28
out portd, r28    ;скопировали содержимое r28 в portd
rcall Main3       ;переходим на подпрограмму задержки
main1:
clr   R28             ;очистили R28
ldi  R28,0b01101101   ;загрузили 0b01101101 в R28
out portd, r28        ;скопировали содержимое r28 в portd
rcall Main3           ;переходим на подпрограмму задержки
main2:
clr   R28         ;очистили R28
ldi  R28,0b00111101;загрузили 0b00111101 в R28
```

```
out portd, r28     ;скопировали содержимое r28 в portd
rcall Main3        ;переходим на подпрограмму задержки
main4:
clr   R19              ;очистили R28
ldi  R28,0b00011011    ;загрузили 0b00011011 в R28
out portd, r28         ;скопировали содержимое r28 в portd
rcall Main3            ;переходим на подпрограмму задержки
main5:
clr   R28         ;очистили R28
ldi  R28,0b00110111;загрузили 0b00110111 в R28
out portd, r28     ;скопировали содержимое r28 в portd
rcall Main3        ;переходим на подпрограмму задержки
main6:
clr   R28         ;очистили R28
ldi  R28,0b01110111;загрузили 0b01110111 в R28
out portd, r28     ;скопировали содержимое r28 в portd
rcall Main3        ;переходим на подпрограмму задержки
main7:
clr   R28         ;очистили R28
ldi  R28,0b00011100;загрузили 0b00011100 в R28
out portd, r28        ;скопировали содержимое r28 в portd
rcall Main3        ;переходим на подпрограмму задержки
main8:
clr   R28              ;очистили R28
ldi  R28,0b01111111    ;загрузили 0b01111111 в R28
out portd, r28         ;скопировали содержимое r28 в portd
rcall Main3            ;переходим на подпрограмму задержки
main9:
clr   R28              ;очистили R28
ldi  R28,0b00111111    ;загрузили 0b00111111 в R28
out portd, r28         ;скопировали содержимое r28 в portd
rcall Main3            ;переходим на подпрограмму задержки
main3:          ldi r23, 200                        ; Загрузить в
step1:  ldi r24, 200                        ; регистры
step2:  ldi r25, 10; константы.
```

```
step3:  dec   r25                    ; Уменьшить на 1 содержимое регистра и
brne step3                    ; если в нем не 0 перейти на метку.
dec   r24
brne step2
dec   r23
brne step1
ldi   R28,0b00000000    ;загрузили 0b00000000 в R28
out   PORTD,R28         ;скопировали содержимое r28 в portd
rjmp REID               ;перейти в главную программу
PRER:
ldi r23, 57                   ; Загрузить в
step4:  ldi r24, 57                   ; регистры
step5:  ldi r25, 10       ; константы.
step6:  dec   r25                    ; Уменьшить на 1 содержимое регистра и
brne step6                    ; если в нем не 0 перейти на метку.
dec   r24
brne step5
dec   r23
brne step4
ret                     ;вернутся из прерывания
Maini:
sbis PINB, $0 ;Подпрограмма пропускания если бит в регистре PINB0=0 не установлен.
rcall Prover ;Переход на подпрограмму реализующую задержку и горения светодиода при нажатой 1 кнопке (т.е PIND5=0)
RJMP Maini
prover:
rcall pre
rcall bor
bor:
sbis pinb, $0
rcall Prover1
RJMP maini0
prover1:
rcall pre
rcall bor1
```

```
bor1:
sbis pinb, $0
rcall Prover2
RJMP maini1
prover2:
rcall pre
rcall bor2
bor2:
sbis pinb, $0
rcall Prover3
RJMP maini2
prover3:
rcall pre
rcall bor3
bor3:
sbis pinb, $0
rcall Prover4
RJMP maini4
prover4:
rcall pre
rcall bor4
bor4:
sbis pinb, $0
rcall Prover5
RJMP maini5
prover5:
rcall pre
rcall bor5
bor5:
sbis pinb, $0
rcall Prover6
RJMP maini6
prover6:
rcall pre
rcall bor6
```

```
bor6:
sbis pinb, $0
rcall Prover7
RJMP maini7
prover7:
rcall pre
rcall bor7
bor7:
sbis pinb, $0
rcall Prover8
RJMP maini8
prover8:
rcall pre
rcall bor8
bor8:
sbis pinb, $0
rcall maini
RJMP maini9
Maini0:
ldi  R28,0b00011000   ;загрузили 0b00011000 в R28
out portd, r28    ;скопировали содержимое r28 в portd
rcall Maini3       ;переходим на подпрограмму задержки
maini1:
clr   R28             ;очистили R28
ldi  R28,0b01101101   ;загрузили 0b01101101 в R28
out portd, r28        ;скопировали содержимое r28 в portd
rcall Maini3           ;переходим на подпрограмму задержки
maini2:
clr   R28         ;очистили R28
ldi  R28,0b00111101;загрузили 0b00111101 в R28
out portd, r28     ;скопировали содержимое r28 в portd
rcall Maini3        ;переходим на подпрограмму задержки
maini4:
clr   R19              ;очистили R28
ldi  R28,0b00011011    ;загрузили 0b00011011 в R28
```

```
out portd, r28        ;скопировали содержимое r28 в portd
rcall Maini3          ;переходим на подпрограмму задержки
maini5:
clr   R28        ;очистили R28
ldi  R28,0b00110111;загрузили 0b00110111 в R28
out portd, r28     ;скопировали содержимое r28 в portd
rcall Maini3        ;переходим на подпрограмму задержки
maini6:
clr   R28        ;очистили R28
ldi  R28,0b01110111;загрузили 0b01110111 в R28
out portd, r28     ;скопировали содержимое r28 в portd
rcall Maini3        ;переходим на подпрограмму задержки
maini7:
clr   R28        ;очистили R28
ldi  R28,0b00011100;загрузили 0b00011100 в R28
out portd, r28        ;скопировали содержимое r28 в portd
rcall Maini3        ;переходим на подпрограмму задержки
maini8:
clr   R28             ;очистили R28
ldi  R28,0b01111111    ;загрузили 0b01111111 в R28
out portd, r28        ;скопировали содержимое r28 в portd
rcall Maini3          ;переходим на подпрограмму задержки
maini9:
clr   R28             ;очистили R28
ldi  R28,0b00111111    ;загрузили 0b00111111 в R28
out portd, r28        ;скопировали содержимое r28 в portd
rcall Maini3          ;переходим на подпрограмму задержки
maini3:
ldi r23, 100                    ; Загрузить в
step7:  ldi r24, 100                    ; регистры
step8:  ldi r25, 10; константы.
step9:  dec   r25                       ; Уменьшить на 1 содержимое регистра и
brne step9                      ; если в нем не 0 перейти на метку.
dec   r24
brne step8
```

```
dec   r23
brne step7
ldi   R28,0b00000000    ;загрузили 0b00000000 в R28
out   PORTD,R28         ;скопировали содержимое r28 в portd
rjmp main               ;перейти в главную программу
PRE:
ldi r23, 74                     ; Загрузить в
step10:ldi r24, 73                      ; регистры
step11:ldi r25, 10      ; константы.
step12:dec   r25                        ; Уменьшить на 1 содержимое регистра и
brne step12                     ; если в нем не 0 перейти на метку.
dec   r24
brne step11
dec   r23
brne step10
ret                     ;вернуться из прерывания
```

ПРИЛОЖЕНИЕ В (СПРАВОЧНОЕ)

Характеристики микроконтроллера Atmega8

Отличительные особенности:

а) 8-разрядный высокопроизводительный AVR микроконтроллер с малым потреблением;

б) Прогрессивная RISC архитектура: 130 высокопроизводительных команд, большинство команд выполняется за один тактовый цикл, 32 8-разрядных рабочих регистра общего назначения, полностью статическая работа, приближающаяся к 16 MIPS (при тактовой частоте 16 МГц) производительность, встроенный 2-цикловый перемножитель;

г) Энергонезависимая память программ и данных: 8 Кбайт внутрисистемно программируемой Flash памяти (In-System Self-Programmable Flash), обеспечивает 1000 циклов стирания/записи, дополнительный сектор загрузочных кодов с независимыми битами блокировки, обеспечен режим одновременного чтения/записи (Read-While-Write), 512 байт EEPROM, обеспечивает 100000 циклов стирания/записи, 1 Кбайт встроенной SRAM, программируемая блокировка, обеспечивающая защиту программных средств пользователя;

г) Встроенная периферия: два 8-разрядных таймера/счетчика с отдельным предварительным делителем, один с режимом сравнения, один 16-разрядный таймер/счетчик с отдельным предварительным делителем и режимами захвата и сравнения, счетчик реального времени с отдельным генератором, три канала PWM , 8-канальный аналого-цифровой преобразователь (в корпусах TQFP и MLF), 6 каналов с 10-разрядной точностью, 2 канала с 8-разрядной точностью, 6-канальный аналого-цифровой преобразователь (в корпусе PDIP), 4 канала с 10-разрядной точностью, 2 канала с 8-разрядной точностью, байт-ориентированный 2-проводный последовательный интерфейс, программируемый последовательный USART, последовательный интерфейс SPI (ведущий/ведомый), программируемый сторожевой таймер с отдельным встроенным генератором, встроенный аналоговый компаратор;

д) Специальные микроконтроллерные функции: сброс по подаче питания и программируемый детектор кратковременного снижения напряжения питания, встроенный калиброванный RC-генератор, внутренние и внешние источники прерываний, пять режимов пониженного потребления: Idle, Power-save, Power-down, Standby и снижения шумов ADC;

е) Выводы I/O и корпуса: 23 программируемые линии ввода/вывода, 28-выводной корпус PDIP, 32-выводной корпус TQFP и 32-выводной корпус MLF;

ж) Рабочие напряжение: 4,5 - 5,5 В;

з) Рабочая частота: 0 - 16 МГц (ATmega8).

ПРИЛОЖЕНИЕ Г (СПРАВОЧНОЕ)

Характеристики микроконтроллера Atmega8535

Отличительные особенности:

а) Высокая производительность при малом потреблении.

б) Развитая RISC архитектура: (130- команд, большинство исполняемых за один машинный такт, 32x8 рабочих регистра общего назначения, полностью статический режим, до 16 MIPS производительность при 16 МГц, встроенный 2-х тактный умножитель.

в) Энергонезависимая память программ и память данных: 8К байтов FLASH памяти программ с внутрисистемным самопрограммированием, 10000 циклов записи- стирания, загрузочная область памяти с независимыми ключевыми битами, 512 байтов EEPROM, 100000 циклов записи- стирания, 512 байтов внутренней SRAM, программируемый ключ доступа к программам и памяти данных.

г) Периферия: два 8- битных таймера/счётчика с программируемым предделителем и режимом сравнения, один 16 битный таймер/ счётчик с программируемым предделителем, режимом сравнения и захвата, счётчик реального времени с программируемым генератором, четыре ШИМ генератора 8 канальный, 10- битный АЦП 8 простых униполярных входа, 7 дифференциальных входа (только для TQFP), 2 дифференциальных входа с программируемым усилением (x1, x10, x200) (только для TQFP), байт ориентированный двухпроводный интерфейс, программированный последовательный USART, Master/Slave SPI, встроенный аналоговый компаратор, программируемый Watchdog с переключаемыми генераторами, специальные функции контроллера. Reset при включении и понижении напряжения питания, внешние и внутренние источники прерывания, внутренний калиброванный RC генератор, шесть режимов энергосбережения: Idle, подавление шума АЦП, экономичный, выкл., ожидания и расширенный режим ожидания.

д) 40- выводной корпус PDIP, 44-выводной TQFP, 44-х PLCC, и 44-х MLF.

е) 32 программируемых входа-выхода.

ж) Рабочее напряжение питания: 4.5 В до 5.5 В ATmega8535.

з) Рабочая тактовая частота: 0-16 МГЦ ATmega8535 [13].

ПРИЛОЖЕНИЕ Д (СПРАВОЧНОЕ)

Конструкторская документация передающей части ПДУ

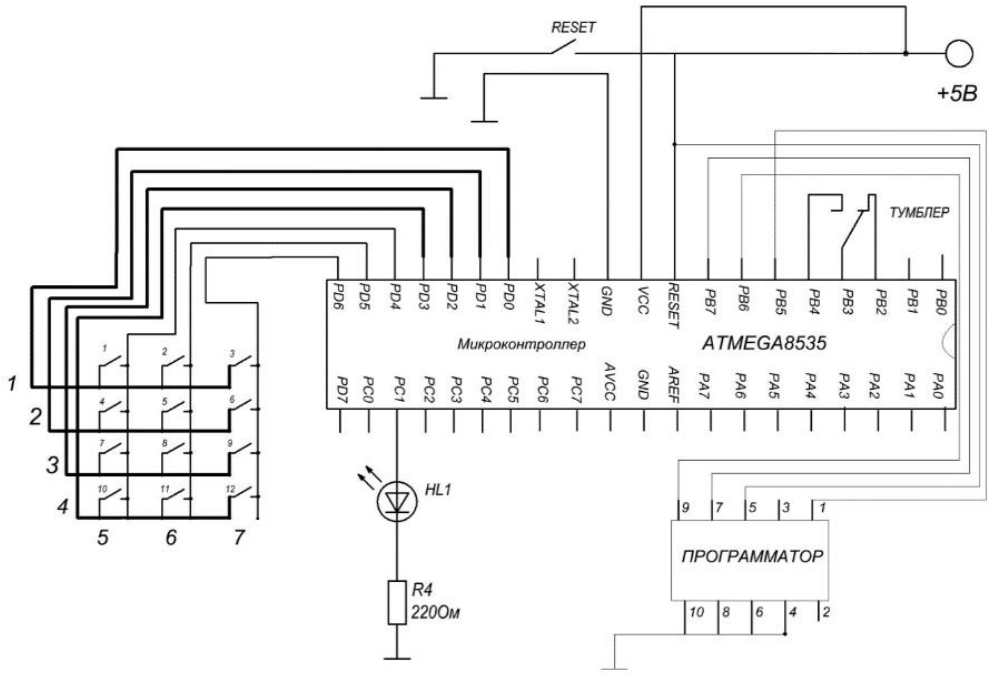

					Схема электрическая принципиальная		
Изм	Лист	№ докум.	Подпис	Дат	Передающая часть пульта дистанционного управления	Лит.	Масса	Масштаб
Разраб.		Ни В.В.					1	1 : 1
Провер.		Кочеткова						
Т. Контр.						Лис 1	Листов 6	
Рецензиз.						РФФ 738		
Н. Контр.								
Утверд.								

Поз. обозн.	Наименование	Кол.	Примечание
Элем.	Наименование	Кл.	Прим.
R4	СП5-2 – 220 Ом ±5ГОСТ В 20.39.404-81	1	
	Микроконтроллер Atmega8535	1	
	Клавиатура AK-304FM-N-WWB	1	
HL1	Светодиод L-34F3C		
Reset	Кнопка	1	
	Тумблер	1	

Изм.	Лист	№ докум.	Подпис	Дат	Перечень элементов			
Разраб.		Ни В.В.			Передающая часть пульта дистанционного управления	Лит.	Лист	Листов
Провер.		Кочеткова					2	6
Реценз.								
Н. Контр.								
Утверд.								

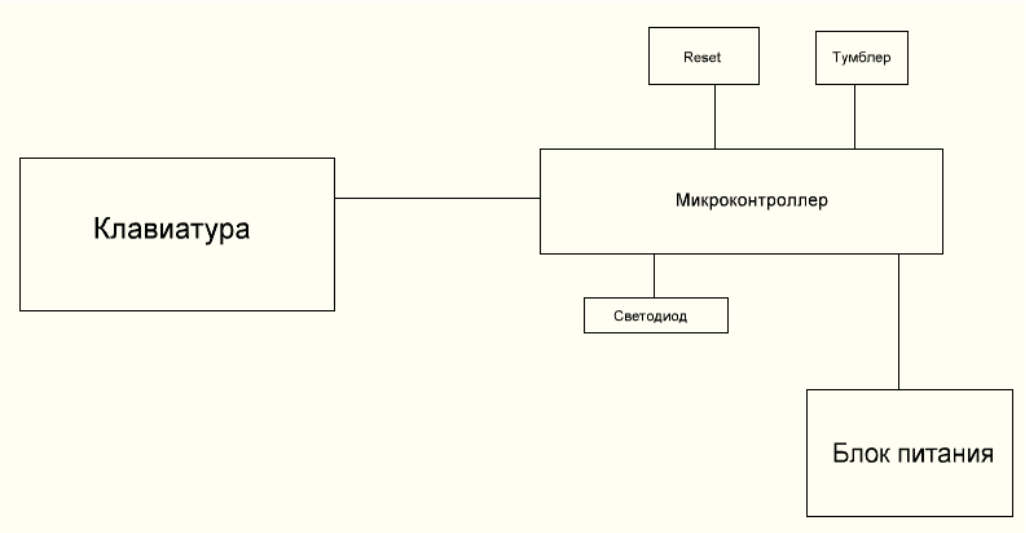

					Схема подключения			
Изм.	Лист	№ докум.	Подпис	Дат	Передающая часть пульта дистанционного управления	Лит.	Масса	Масштаб
Разраб.		Ни В.В.					1	1 : 1
Провер.		Кочеткова						
Т. Контр.						Лис 3	Листов 6	
Рецензиз.						РФФ 738		
Н. Контр.								
Утверд.								

ПРИЛОЖЕНИЕ Е (СПРАВОЧНОЕ)

Конструкторская документация приемной части ПДУ

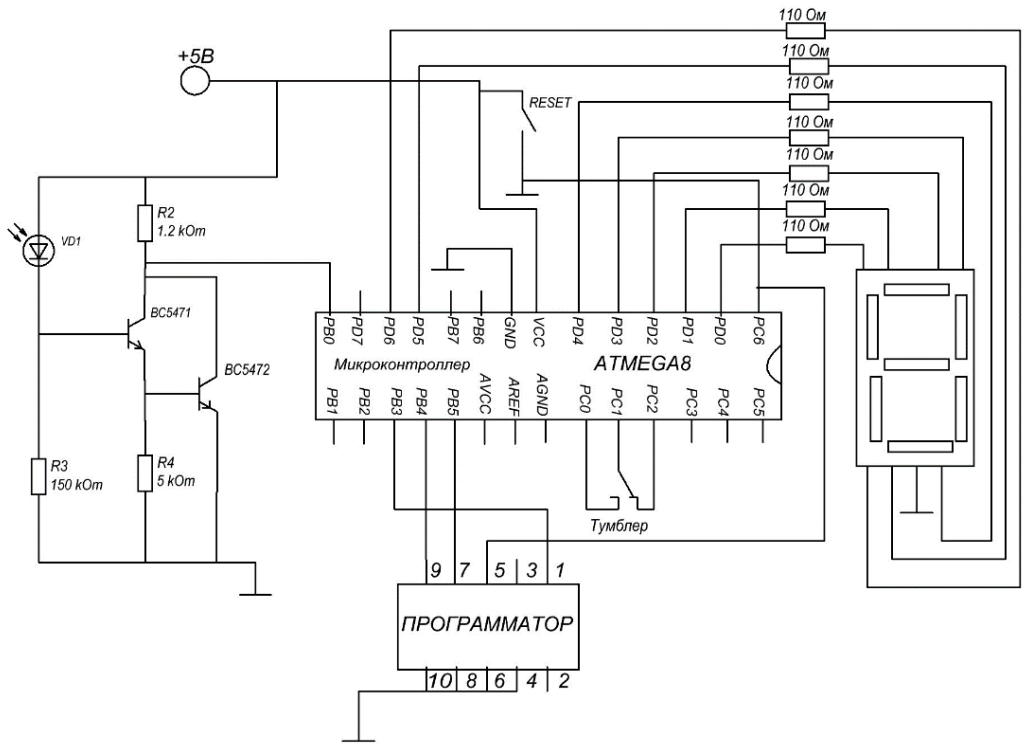

					Схема электрическая принципиальная		
Изм	Лист	№ докум.	Подпис	Дат	Приемная часть пульта дистанционного управления	Лит.	Масса	Масштаб
Разраб.		Ни В.В.					1	1 : 1
Провер.		Кочеткова				Лис 4	Листов 6	
Т. Контр.								
Рецензиз.						РФФ 738		
Н. Контр.								
Утверд.								

Поз. обозн.	Наименование	Кол.	Примечание
Элем.	Наименование	Кл.	Прим.
	С2-23 – 110 Ом	7	
R2	СП5-2 – 1.2 кОм	1	
R3	СП5-2 – 150 кОм		
R4	СП5-2 – 5 кОм	1	
BC547	BC5471		
Reset	Кнопка	1	
VD1	Фотодиод BL-L512PD	1	
	Семисегментный индикатор	1	
	Тумблер	1	

Изм.	Лист	№ докум.	Подпис	Дат	Перечень элементов			
Разраб.		Ни В.В.			Приемная часть пульта дистанционного управления	Лит.	Лист	Листов
Провер.		Кочеткова					5	6
Реценз.								
Н. Контр.								
Утверд.								

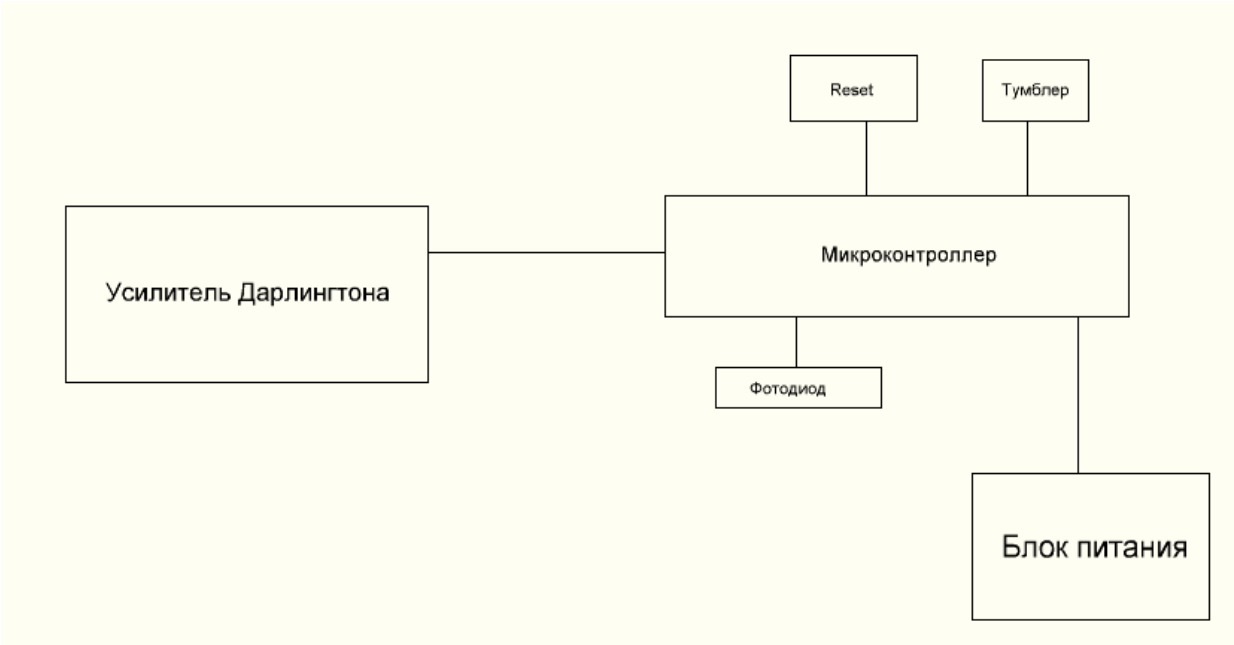

					Схема подключения			
Изм.	Лист	№ докум.	Подпис	Дат	Приемная часть пульта дистанционного управления	Лит.	Масса	Масштаб
Разраб.		Ни В.В.					1	1 : 1
Провер.		Кочеткова						
Т. Контр.						Лис 6	Листов 6	
Реценз.						РФФ 738		
Н. Контр.								
Утверд.								

Отчет о проверке на заимствования №1

Автор: Репин Александр repinalexander95@bk.ru / ID: 4981069
Проверяющий: Репин Александр (repinalexander95@bk.ru / ID: 4981069)
Отчет предоставлен сервисом «Антиплагиат»- http://users.antiplagiat.ru

Информация о документе

№ документа: 2
Начало загрузки: 14.02.2019 18:35:47
Длительность загрузки: 00:00:01
Имя исходного файла: пробная версия
Размер текста: 1737 кБ
Символов в тексте: 79281
Слов в тексте: 10448
Число предложений: 455

Информация об отчете

Последний готовый отчет (ред.)
Начало проверки: 14.02.2019 18:35:49
Длительность проверки: 00:00:03
Комментарии: не указано
Модули поиска:

Заимствования

13,1%

Цитирования

0%

Оригинальность

86,9%

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.

Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.

Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.

Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.

Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.

Заимствования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.

Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Доля в тексте	Источник	Ссылка	Актуален на	Модуль поиска	Блоков в отчете	Блоков в тексте
[01]	2,64%	3,59%	793	http://docme.ru	07 Мая 2017	Модуль поиска Интернет	33	34
[02]	1,41%	3,02%	►►►-►►►► ►►►