МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева (Самарский университет)»

Институт информатики и кибернетики

Кафедра информационных систем и технологий

КУРСОВОЙ ПРОЕКТ ПО ДИСЦИПЛИНЕ

«Микропроцессорные средства и системы»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

«Проектирование микропроцессорных устройств на базе однокристальных микроконтроллеров»

|  |  |
| --- | --- |
| Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись)* | Н.А. Бурмистров |
|  |  |
| Руководитель проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись)* | В.Г. Иоффе |
|  |  |

САМАРА 2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования «Самарский национальный исследовательский   
университет имени академика С.П. Королева»

(Самарский университет)

Институт информатики и кибернетики

Кафедра информационных систем и технологий

ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ

Обучающемуся Бурмистрову Никите Александровичу группы 6403-090301D

Тема проекта: ***«***Проектирование микропроцессорных устройств на базе однокристальных микроконтроллеров» в соответствии заданием:

Разработать контроллер, реализующий следующие функции: опрос состояния матрицы с 15 клавишами, включение соответствующего канала АЦП (номер канала определяется номером контакта), разрядность которого 10 бит, быстродействие – не менее 100 мкс, входное напряжение не более 2,56 В. В каждом канале необходимо контролировать Nmini≤Ni≤Nmaxi, где Nmini, Nmaxi – соответственно минимальное и максимальное значение i-того канала, а Ni –текущее значение. Nmini, Nmaxi хранятся в ПЗУ. При выходе за указанные пределы включить соответствующие светодиоды. Окончание измерения индицируется сигналом (отрицательный импульс 10 мкс).

По внешнему запросу прерывания данные передать в приемник через интерфейс RS-232 на скорости 19200 б/с. Данные передаются в следующем формате – номер канала, результат измерения в двоичной форме. Если результат измерения не соответствует заданному диапазону, то вместо значения передать код $С000.

|  |  |  |
| --- | --- | --- |
| Планируемые результаты освоения образовательной программы (компетенции) | Планируемые результаты курсового проектирования | Содержание задания |
| ПК-1 Способен обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности  ПК-1.4. Проектирует информационные системы, разрабатывает программу работ по экспериментальной проверке ее эффективности | Знать: технические характеристики микроконтроллеров и инструментальных средств.  Уметь: проектировать устройства на микроконтроллерах.  Владеть: навыками  решения задач интегрирования в систему микропроцессорных устройств и их комплексной отладки с целью повышения эффективности | 1.Анализ полученного задания  2. Разработка структурной организации  3.Анализ алгоритмов обработки и управления  4.Разработка принципиальной электрической схемы  5.Выбор инструментальных средств и языка программирования  6. Разработка программы  7.Тестирование разработанного контроллера  8. Разработка инструкции по эксплуатации  9. Заключение |

Дата выдачи задания 5 сентября 2022 г.

Срок представления на кафедру пояснительной записки 28 декабря 2022 г.

Руководитель курсового проекта

доцент каф. ИСТ к.т.н. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.Г. Иоффе

*(подпись)*

Задание принял к исполнению

обучающийся группы № 6403-090301D \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.А. Бурмистров

*(подпись)*

РЕФЕРАТ

Пояснительная записка к курсовому проекту: 35 c., 17 рисунков, 2 таблицами, 3 источника, 3 приложения.

МИКРОКОНТРОЛЛЕР, КЛАВИАТУРА, СВЕТОДИОД, ПРИЕМОПЕРЕДАТЧИК RS-232, ИЗМЕРЕНИЕ НАПРЯЖЕНИЯ.

В соответствии с техническим заданием спроектирована система на основе ОМК ATmega8535 семейства AVR фирмы Atmel, предназначенная для измерения напряжения с каналов, сигнализации об выходе измеряемого напряжения за пределы минимальных и максимальных значений, передачи данных через интерфейс RS232. Разработаны алгоритмы функционирования, структурная схема, а также программное обеспечение.

Система реализована и отлажена средствами Proteus 8 Professional.

Выполнена отладка и проверка системы на работоспособность.

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧНИЙ, ЕДИНИЦ И ТЕРМИНОВ

МПС – микропроцессорная система;

МПУ – микропроцессорное устройство;

ОМК – однокристальный микроконтроллер;

ПБ – процессорный блок;

ПЗУ (ROM) – постоянное запоминающее устройство;

ОЗУ (RAM) – оперативно запоминающее устройство;

ГСА – граф-схема алгоритма;

Рг – регистр;

ШД – шина данных;

КГ – кварцевый генератор;

ИОН – источник опорного напряжения

СТ – счетчик таймер.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 7](#_Toc122719429)

[1 СТРУКТУРНАЯ ОРГАНИЗАЦИЯ 8](#_Toc122719430)

[2 АЛГОРИТМЫ ОБРАБОТКИ И УПРАВЛЕНИЯ 12](#_Toc122719431)

[3 ВЫБОР ЭЛЕМЕНТОЙ БАЗЫ 17](#_Toc122719432)

[3.1 Выбор ОМК 17](#_Toc122719433)

[3.2 Выбор приемопередатчика 18](#_Toc122719434)

[3.3 Выбор мультиплексора 18](#_Toc122719435)

[3.4 Выбор ИОН 19](#_Toc122719436)

[4. ПРИНЦИПИАЛЬНАЯ ЭЛЕКТРИЧЕСКАЯ СХЕМА 20](#_Toc122719437)

[5 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ 22](#_Toc122719438)

[6 ИНСТРУКЦИЯ ПО ЭКСПЛУАТАЦИИ КОНТРОЛЛЕРА 23](#_Toc122719439)

[7 АЛГОРИТМ ТЕСТИРОВАНИЯ 30](#_Toc122719440)

[ЗАКЛЮЧЕНИЕ 33](#_Toc122719441)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 34](#_Toc122719442)

[ПРИЛОЖЕНИЕ А – Принципиальная схема 35](#_Toc122719443)

[ПРИЛОЖЕНИЕ Б – Спецификация элементов 36](#_Toc122719444)

[ПРИЛОЖЕНИЕ В – Листинг программы 37](#_Toc122719445)

# ВВЕДЕНИЕ

В рамках курсового проекта необходимо реализовать систему, позволяющую измерять напряжение с одного из 15 каналов по нажатию на соответствующую кнопку на клавиатуре, с последующим выводом результата на терминал и формированием сигнала при завершении измерения.

Для решения поставленной задачи эффективно использовать МПС, реализуемую на базе однокристального микроконтроллера (ОМК).

ОМК обладают многими преимуществами – небольшая стоимость и энергопотребление, а также высоким функционалом, достаточным для поставленной задачи [1]. ОМК могут содержать в своем составе порты ввода/вывода, аналого-цифровые и цифро-аналоговые преобразователи, счетчики-таймеры, аналоговые компараторы и другие периферийные устройства, что избавляет от необходимости их отдельной покупки. Высокая скорость обработки данных позволяет использовать их для решения инженерных задач широкого спектра. В то время как разработка МПС на базе микропроцессоров требует применения большого числа микросхем различного назначения, что существенно увеличивает стоимость, габариты, энергопотребление и снижает надежность контроллера [2].

# 1 СТРУКТУРНАЯ ОРГАНИЗАЦИЯ

Для рассмотренного выше задания необходимо ПБ, ПЗУ, ОЗУ, блок обработки прерываний, счетчики, индикаторы, мультиплексор, регистры ввода/вывода, интерфейс RS-232.

Для измерения напряжения с определенного канала требуется матричная клавиатура, вмещающая в себе 15 кнопок, по нажатию на которые принимается напряжение в АЦП на соответствующем канале. Для постоянного опроса матричной клавиатуры потребуется счетчик таймер и блок прерываний.

Для измерения напряжений с 8 каналов и более потребуется мультиплексор. Управление мультиплексором может осуществляться с помощью восьмиразрядного регистра выхода.

Управление светодиодами может выполняться восьмиразрядным выходным регистром.

Передача информации по интерфейсу RS-232 может быть реализована асинхронным приемопередатчиком UART(USART) и преобразователем сигнала MAX-232.

Практически требуемый объем ОЗУ и ПЗУ может быть определен экспериментально с помощью симулятора после выбора ОМК и разработки программного обеспечения. Тогда можно выбрать новую конфигурацию ОМК с целью сокращения его стоимости.

Для обработки внешних управляющих сигналов контроллер должен включать в себя блок обработки прерываний.

Объем ПЗУ требуемый для хранения констант может быть определен на основании следующих расчетов:

- хранение минимальных значений напряжений на каналах – 15x8 = 120 байт;

- хранение максимальных значений напряжений на каналах – 15x8 = 120 байт;

- хранение измерений напряжений с каналов– 15x8 = 120 байт;

Параметры процессорного блока ПБ определяются особенностями алгоритмов обработки и требуемым быстродействием.

Структура контроллера для рассмотренного примера представлена на рисунке 1.

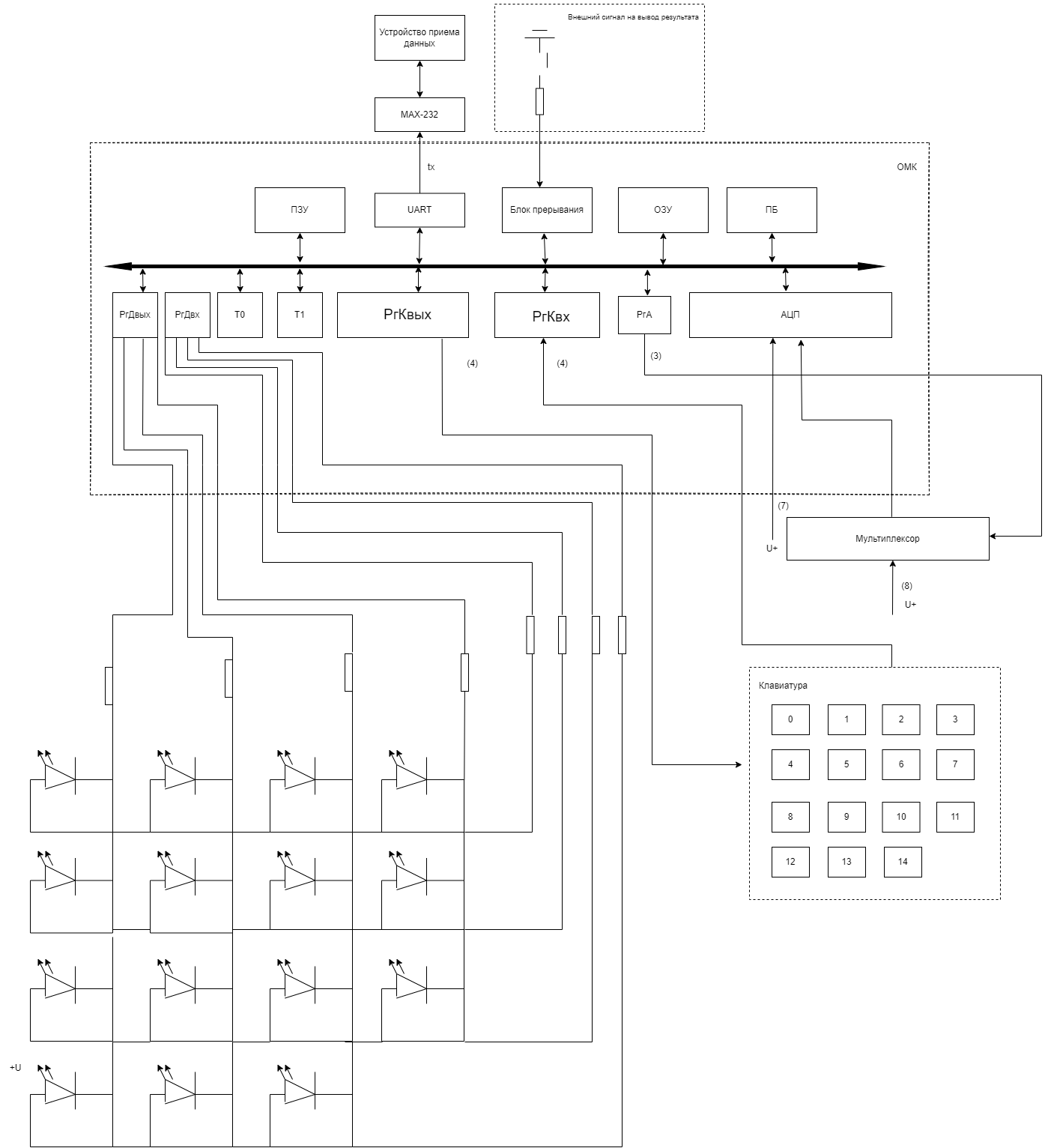


Рисунок 1 – Структурная схема контроллера

Схема состоит из следующих элементов:

* ПБ (процессорный блок) ­­–­­­­­­­ выполняет основные функции по обработку и управлению данными;
* ОЗУ (оперативное запоминающее устройство) – устройство, хранящее входные и выходные данные, а также промежуточные результаты;
* ПЗУ (постоянное запоминающее устройство) – устройство, хранящее программы обработки данных и запуска устройства.
* Блок обработки прерывания – элемент, который позволяет прервать обработку программы для выполнения подпрограммы;
* T0 (счетчик таймер) – устройство, предназначенное для опроса клавиатуры по переполнению;
* T1 (счетчик таймер) – устройство, предназначенное для генерации прерывания по захвату;
* РгДВых (регистр выхода диодов) – регистр, предназначенный для генерации выходных сигналов на матрицу диодов ­
* РгДВх (регистр входа диодов) – регистр, предназначенный программной реализации включения диодной матрицы
* Устройство приема данных ­ – устройство, которое принимает данные по MAX232
* USART(UART) – последовательный порт, через который с помощью MAX232 осуществляется отправка данных на устройство приема данных
* Мультиплексор – устройство, позволяющая увеличить количество линий для измерения на АЦП
* РгКвх (регистр входа клавиатуры) – регистр, получить адрес нажатой клавиши на клавиатуре
* РгКвых (регистр выхода клавиатуры) – регистр, позволяющий программно реализовать опрос клавиатуры
* РгА (регистр адреса) – регистр, позволяющий указывать мультиплексору пропускаемый канал на измерение
* Клавиатура – устройство, предназначенное для выбора нужного канала для измерения на нем напряжения.

# 2 АЛГОРИТМЫ ОБРАБОТКИ И УПРАВЛЕНИЯ

Разработка алгоритма функционирования МПУ выполняется на основе технического задания и структурной схемы. Одним из наиболее распространенных средств разработки алгоритмов является проектирование на основе обобщенной граф-схемы алгоритма ГСА.

Граф-схема алгоритма представляется совокупностью последовательно исполняемых операторов: так называемых операторных отображающих преобразование данных, и условных вершин, отражающих проверки входов и результатов исполнения предыдущих шагов с целью выбора пути продолжения исполнения.

Алгоритм, работающий в соответствии с заданием по данному курсовому проекту, выглядит следующим образом:

1. Указание диапазона между минимальным и максимальным напряжениями для каждого из 15 каналов;
2. Инициализация всех блоков микроконтроллера;
3. Опрос клавиатуры с 15 клавишами;
4. Измерение напряжение на соответствующем канале;
5. Подача отрицательного импульса по завершению измерения;
6. При выходе за диапазон измерения включить соответствующий светодиод;
7. По сигналу передать значение через интерфейс RS-232;

На рисунке 2 представлена обобщенная граф-схема алгоритма.



Рисунок 2 – Обобщённая граф схема алгоритма

Алгоритм «бегущий ноль» реализован следующим образом:

1. Обход матрицы по ширине и подача сигнала на соответствующую колонку
2. Проверка нажатия клавиши в колонке
3. Установление индекса в переменную, соответствующего клавише

На рисунке 3 представлена граф-схема алгоритма получения индекса нажатой клавиши.



Рисунок 3 – Алгоритм опроса клавиатуры

На схеме использовались обозначения:

width – ширина матричной клавиатуры;

length – высота матричной клавиатуры;

i – вспомогательная переменная для нахождения индекса нажатой кнопки;

j – вспомогательная переменная для нахождения индекса нажатой кнопки;

channelNumber – номер нажатой клавиши;

canReadChannel – была ли нажата клавиша на клавиатуре;

Разрядность данных:

i [7:0], j[7:0], channelNumber[7:0], canReadChannel[1:0].

Алгоритм передачи данных по интерфейсу USART реализован следующим образом:

1. Запись данных в переменную data;
2. Ожидание возможности записи данных в сдвиговый регистр;
3. Передача значения из data в сдвиговый регистр;
4. Ожидание завершения передачи ;

На рисунке 4 представлена граф-схема алгоритма передача данных по USART.



Рисунок 4 – Алгоритм передачи данных по USART

На схеме использовались обозначения:

data – передаваемые данные.

Разрядность данных:

data [7:0].

# 3 ВЫБОР ЭЛЕМЕНТОЙ БАЗЫ

## 3.1 Выбор ОМК

Выбранный в соответствии с техническим заданием ОМК должен обладать функциональной полнотой, чтобы минимизировать число внешних компонентов, необходимых для реализации требуемого МПУ, иметь минимальное количество неиспользуемых линий ввода-вывода, минимальное энергопотребление и стоимость.

По заданию требуется подобрать ОМК, в состав которого должны входить следующие элементы:

* ПБ;
* ОЗУ;
* ПЗУ;
* 31 линия ввода-вывода
* 10 разрядный АЦП
* Интерфейс USART
* Блок управления прерывания

Для данного проекта выбран ОМК ATMEGA8535, который обладает следующими параметрами:

* тактовой частотой до 16 МГц;
* объемом памяти программ 8 кбайт;
* наличием EEPROM 512 байт;
* наличием встроенного интерфейса USART;
* напряжением питания в диапазоне от 4.5 до 5.5 В;
* количеством 32 выходов/входов.
* наличием 16 битного таймера;
* наличием INT0, INT1 и INT2.

Данный ОМК соответствует требованиям технического задания, обладает функциональной полнотой, минимизирует число внешних компонентов, необходимых для реализации МПУ.

Таблица 1 – Распределение линий ОМК

|  |  |
| --- | --- |
| Наименование линии | Назначение |
| PC0 – PC7 | Линии подключения клавиатуры |
| PA0 – PA7 | Входы АЦП |
| PB0 – PB7 | Линии подключения светодиодов |
| PD0 – PD1 | Линии USART |
| RESET | Линия сброса |
| PD2 | Вход внешнего запроса прерывания |
| PD4 | Выход отрицательно импульса |
| PD5-PD7 | Линии управления мультиплексором |

Настройки периферийных устройств:

UART – формат обмена данных 8 бит и бит паритета, режим работы асинхронный, скорость передачи данных 19200 бит/c, способ вывода – программный.

## 3.2 Выбор приемопередатчика

Приемопередатчик должен осуществлять прием данных по интерфейсу UART и передавать полученные данные по интерфейсу RS-232. Таким приемником является MAX-232.

Режим работы данного приёмопередатчика является дуплекс, напряжение питания находится в диапазоне от 4.5 до 5.5 В.

## 3.3 Выбор мультиплексора

Одной из основных характеристик при выборе мультиплексора является его разрядность, так как в задании указано, что измерение производится с 15 каналов , но 7 из них идут напрямую в АЦП, то соответственно разрядность мультиплексора должна быть равна 8, чтобы можно было измерить с оставшихся каналов напряжения. Также важно чтобы напряжение работы мультиплексора совпадало с напряжением питания микроконтроллера.

Данным требованиям удовлетворяет восьмиканальный мультиплексор 74HC4051.

## 3.4 Выбор ИОН

Для корректного измерения входного напряжения с помощью АЦП не всегда подходит внутренний источник, поэтому нужно иметь внешний стабильный источник опорного напряжения. По заданию максимальным измеряем напряжением является 2.55 B, следовательно нужно брать источник, который работает в данном диапазоне.

Выберем ИОН AD780, выходное напряжение которого составляет 2,5 или 3,0 B и программируется коммутацией внешних выводов. Погрешность данного источника опорного напряжения составляет ±0.03 %, что указывает на высокую стабильность элемента.

# 4. ПРИНЦИПИАЛЬНАЯ ЭЛЕКТРИЧЕСКАЯ СХЕМА

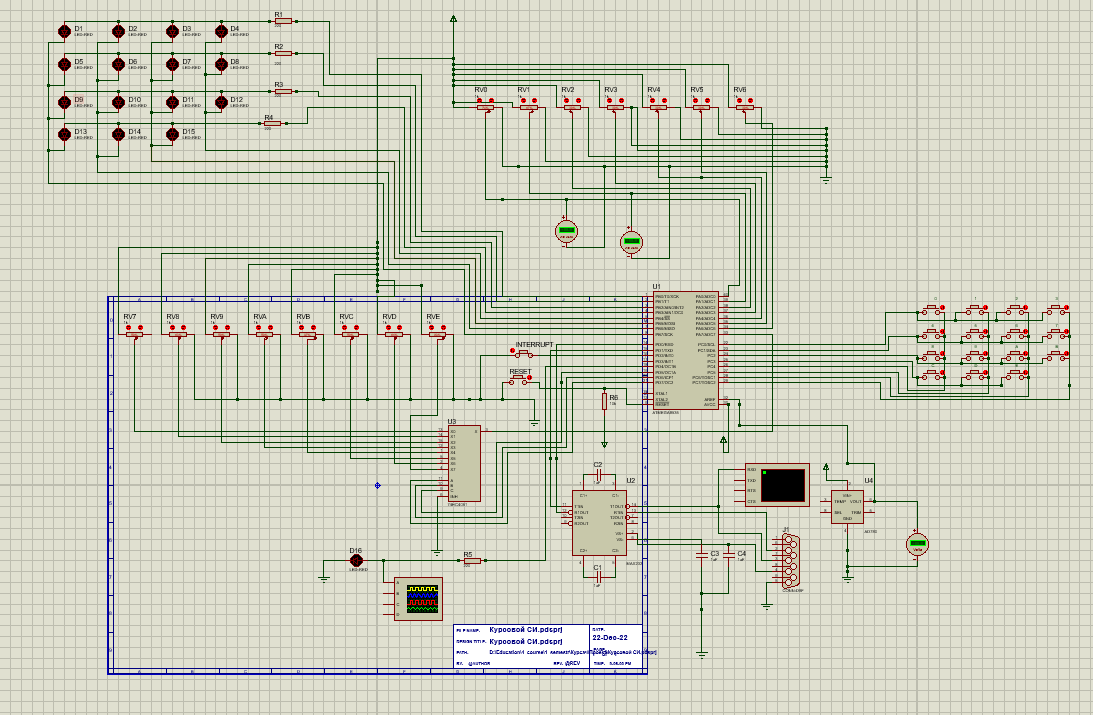


Рисунок 5 – Принципиальная схема

Центральным блоком на схеме является ОМК ATMEGA8535, который работает на частоте 4МГц. Выбор частоты обусловлен высокой частой опроса клавиатуры для того, чтобы ни одно из нажатий не было пропущено.

При включении питания автоматически выполняется сброс микроконтроллера U1, выполненного на микросхеме ATmega8535, что приводит к инициализации всех основных структур программного обеспечения и началу работы микроконтроллера.

В случае некорректного запуска работы контроллера предусмотрена внешняя схема сброса, состоящая из резистора R5 и кнопки RESET.

После инициализации всех устройств начинается опрос матричной клавиатуры, который реализован с помощью алгоритма «бегущий ноль», предназначенный для идентификации нажатой кнопки на клавиатуре.

После нажатия на кнопку на клавиатуре подается сигнал микроконтроллеру с какого канала нужно измерить напряжение, которое регулируется с помощью переменного резистора на каждом из каналов.

Если была нажата клавиша с индексом больше 7, то линиях PD4 – PD7 поддается сигнал на мультиплексор U3 для передачи напряжение с соответствующего канала на PA7 линию приема напряжения АЦП.

По завершению измерения на линию PD4 подается отрицательный сигнал в 10мкс. В случае если результат измерения выходит за пределы между минимальным и максимальным значением, указывающее в программе, то на линии PB0 – PB7 подается напряжение для включение соответствующего светодиода.  
 Для вывода значения нужно подать сигнал микроконтроллеру с помощью нажатия кнопки INTERRUPT, вызывающее программное внутреннее прерывание INT0. Стоит отметить то, что программно реализовано так, что если не было завершено измерение, то прерывание не будет обрабатываться.

Данные передаются по USART через линию PD1 на скорости 19200 бит/c. Так как у ОМК нет аппаратной реализации RS-232, данные передаются по USART, а потом преобразуются в RS-232 посредством интерфейса MAX-232.

# 5 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Для высокой скорости работы и упрощения разработки отлично подходит компилируемый и строго типизируемый язык программирования общего назначения С (Си).

Для разработки программного обеспечения была выбрана платформа Proteus 8 Professional, которая состоит из системы моделирования моделей электронных компонентов. В данной среде отлично разрабатываются системы для работы с микроконтроллерами, микропроцессорами [3].

Данная платформа поддерживает большинство современных семейств процессор, имеет в базе большое количество разнообразных аналоговых элементов и цифровых устройств, поддерживают работу с различными внешними и встроенными компиляторами. Одной из особенностей платформы является удобный графический интерфейс, который значительно ускоряет разработку программного обеспечения.

При разработке программ под МК AVR на СИ, может использоваться любая среда разработки, на выходе которой формируются файлы с расширениями, поддерживаемыми Proteus 8 Professional (HEX, ASM, COF, ELF/DWARF2, UBROF и т.д.).

Proteus 8 отлично подходит для данной курсовой работы по той причине, что находится в бесплатном доступе.

Листинг программы приведен в приложении В.

# 6 ИНСТРУКЦИЯ ПО ЭКСПЛУАТАЦИИ КОНТРОЛЛЕРА

Перед началом измерения нужно указать минимальное и максимальное значения напряжения на линии. (рисунок 6)

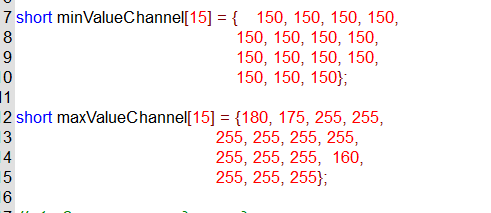


Рисунок 6 – Массивы максимальных и минимальных значений на линии

Дальше нужно выбрать на нужном канале выдаваемое напряжение с помощью регулирования сопротивления на линии (рисунки 7-8)

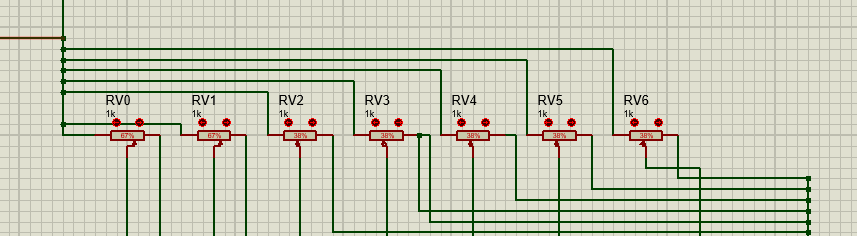


Рисунок 7 – Настройка сопротивлений на линиях 0 - 6

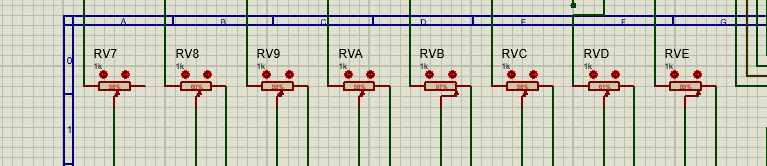


Рисунок 8 – Настройка сопротивлений на линиях 7 - E

После настройки напряжения на линии нажимаем на соответствующую клавишу на клавиатуре для измерения напряжения с помощью АЦП (рисунок 9).

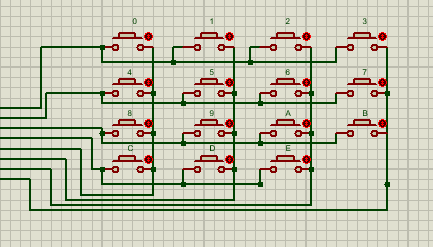


Рисунок 9 – Клавиатура для подачи сигнала на микроконтроллер для измерения напряжения на линии

Для того, чтобы понять завершилось ли измерение напряжения на соответствующем канале, нужно открыть Digital Oscilloscope, в котором завершение измерения индицируется отрицательным сигналом 10 мкс (рисунок 10).

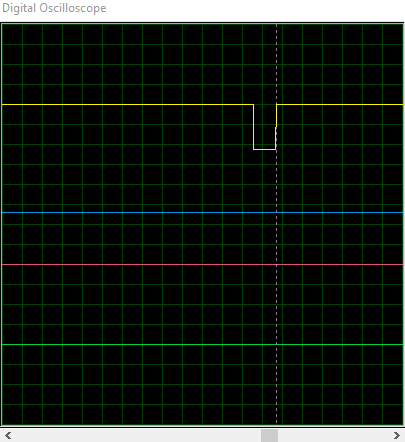


Рисунок 10 – Отрицательный сигнал в 10мкс в осциллографе

В случае если напряжение на выбранном канале выходит за диапазон минимального и максимального значений, то включается соответствующий светодиод (рисунок 11).

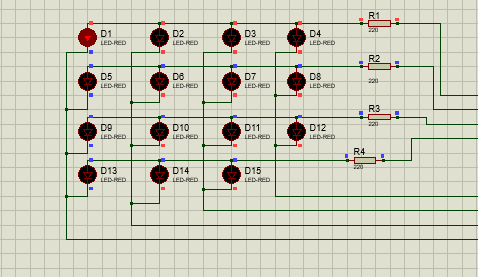


Рисунок 11 – Включение светодиода при выходе за диапазон

Для вывода значений на Virtual Terminal нужно нажать на кнопку INTERRUPT (рисунок 12).

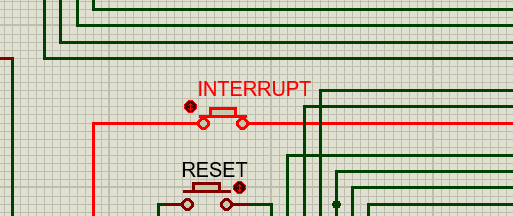


Рисунок 12 – Кнопка для вывода измерений на терминал

Данные выводятся в формате [Номер канала (00-0E)] [Значение напряжения в HEX формате] (рисунок 13).

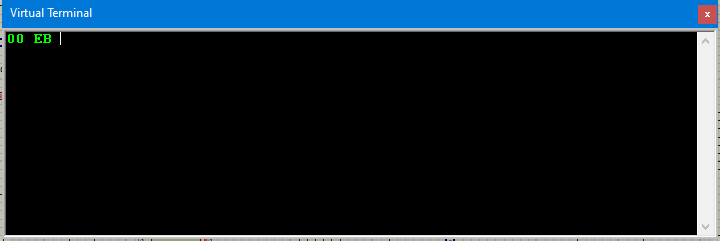


Рисунок 13 – Вывод значения напряжения на терминал

Для перезапуска программы нажмите кнопку RESET.

Тестовый пример 1. Проверим нормальную работу устройства.

Для начала тестирования укажем диапазон напряжений и отобразим их в таблице 2, а также измерить фактическое напряжение на линиях с помощью вольтметров.

Таблица 2 – Диапазоны напряжений на линиях

|  |  |  |  |
| --- | --- | --- | --- |
| Номер канала | Минимальное напряжение, В | Максимальное напряжение, В | Фактическое напряжение, В |
| 1 | 1,5 | 180 | 1,55 |
| 2 | 1,5 | 175 | 1,65 |
| B | 1,5 | 160 | 1,60 |

После нажмем соответствующие клавиши на клавиатуре, которые идентифицируют каналы. Понять то, когда завершилось измерение можно по отрицательную сигналу на линии PD4. Из рисунка 13 видно то, что все значения напряжений каналов попали в указанный диапазон, поэтому ни один из светодиод не включился.

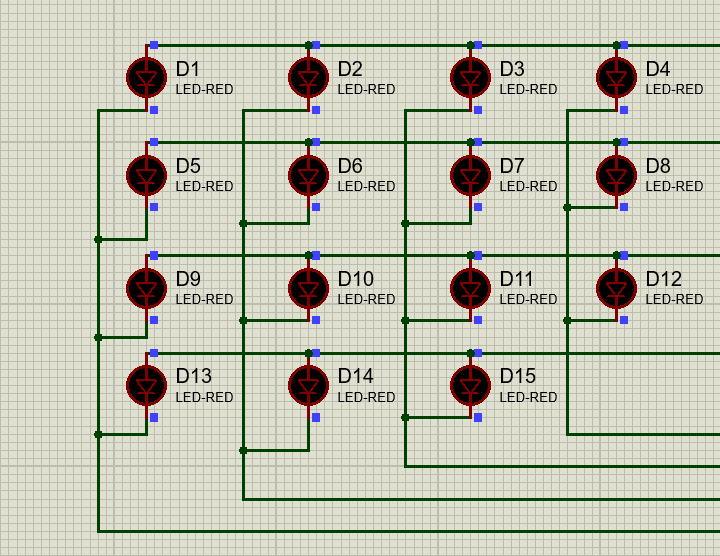


Рисунок 14 – Матрица светодиодов

Вывод результата на терминал после нажатия кнопки INTERRUPT можно наблюдать на рисунке 15.

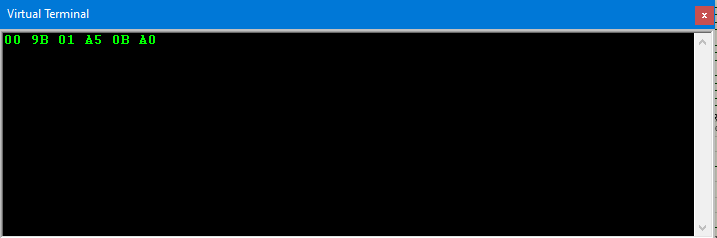


Рисунок 15 – Результат измерения

Результат с фактическим значением на линиях можно наблюдать в таблице 3.

Таблица 3 – Измерение напряжение

|  |  |  |
| --- | --- | --- |
| Номер канала | Фактическое напряжение, В | Вычисленное  напряжение, В |
| 1 | 1,55 | 1,55 |
| 2 | 1,65 | 1,65 |
| B | 1,60 | 1,60 |

Тестовый пример 2. Проверка на измерение, которое выходит за указанный диапазон.

Настроем с помощью регулируемого резистора линию 1 на напряжение в 2 В, а диапазон напряжения в программе укажем от 1,5 В до 1,7 В. После нажмем на кнопку на клавиатуре, которая идентифицирует данный канал. На рисунке 16 можем наблюдать то, как включился светодиод, который символизирует то, что измеряемое напряжение вышло за диапазон допустимого значения.

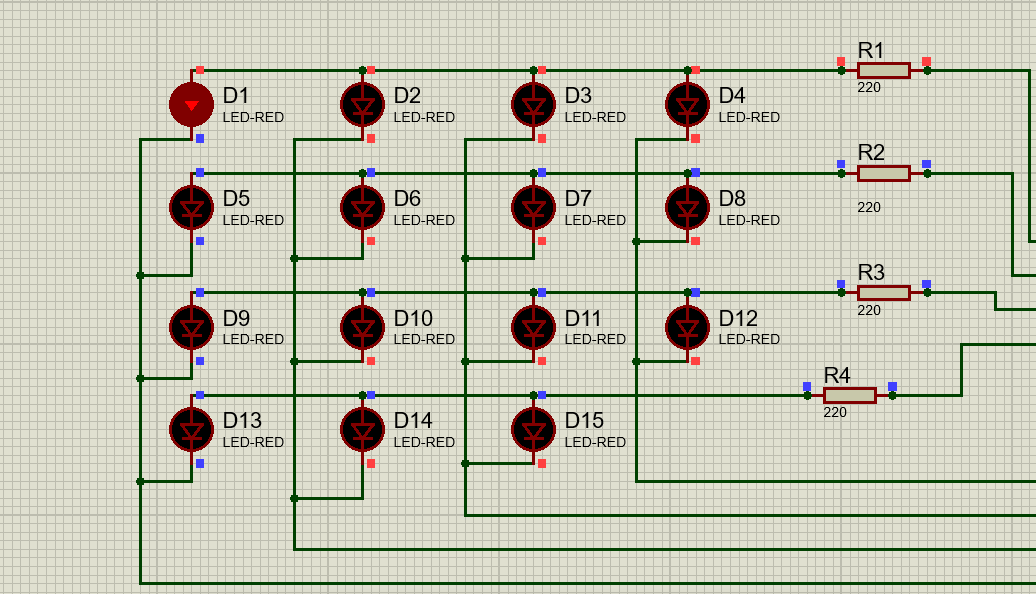


Рисунок 16 – Матрица светодиодов

По нажатию на кнопку INTERRUPT мы можем наблюдать вывод результата, который символизирует нас о том, что измерение вышло за пределы значением C0 00 (рисунок 17)



Рисунок 17 – Результат измерения

# 7 АЛГОРИТМ ТЕСТИРОВАНИЯ

Одной из сложнейший частей разработки программного обеспечения для микроконтроллеров является его тестирование аппаратной и программной части. Тестирование требуется для того, чтобы было понимание того, что программа или аппаратная часть функционирует так, как это было задумано изначально. Тестирование повышают вероятность обнаружение ошибок и ускоряют их устранение.

Тестирование производится при включении МПУ, а также может повторяться с определенной периодичностью. Одной из главных проблем тестирования микроконтроллеров является то, что нельзя сразу понять то, где произошла ошибка – на аппаратной или программной стороне.

В процессе проектирования и отладки МПУ используются  
средства, позволяющие с определенной степенью достоверности  
выполнить тестирование. В основе методов тестирования лежит гипотетическая модель неисправностей, выбираемая на основе практического опыта эксплуатации и настройки данного устройства или системы. Наиболее широкое распространение получил "метод раскрутки", при котором определяется системное ядро, работающее наиболее надежно, а далее к ядру постепенно присоединяются и контролируются остальные компоненты системы (от ядра к периферийным устройствам).

Тестирование ПЗУ

Тестирование ПЗУ производится с помощью метода контрольной суммы, суть которого заключается в том, чтобы сравнивать сумму ячеек с суммой, которая находится в последней ячейке. У такого подхода есть недостаток – кратность ошибок.

Лучшие результаты обеспечивает формирование контрольных циклических избыточных кодов (CRC). Содержимое ПЗУ считывается в 16-разрядный сдвиговый регистр в виде непрерывной последовательности битовых данных. После ввода всех бит ПЗУ результат, зафиксированный в регистре, является CRC.

Программа, тестирующая контрольную сумму, должна включаться вместе с включением питания, сравнивая последнюю ячейку с суммой всех данных, хранящихся в ПЗУ до заданной ячейки. Несовпадение свидетельствует о неисправности ПЗУ. Однако существует вероятность многократной ошибки, которая может дать результат равный байту истинного результата.

Тестирование ОЗУ

Из-за того, что размер ОЗУ является не фиксированным, поэтому нельзя использовать проверку контрольной суммы.

В этом случае чаще всего используют алгоритм под названием «бегущая единица». Суть данного метода заключается в том, что в начале алгоритма все биты в ОЗУ устанавливаются в нулевое значение. Затем справа налево каждый бит устанавливается в единичное значение, а после возвращается в ноль. После смотрятся все биты, в которых была установлена единица и, если фактическое значение не совпадает с тем, что мы произвели, то это сообщает нам о том, что ОЗУ неисправно.

Тестирование USART

Данное тестирование выполняется «эхоконтролем». Требуется участие внешних приемопередатчиков или дополнительных линий портов, на входе/выходе которых программно реализуются требуемые протоколы обмена.

Тестирование АЦП

Для проверки АЦП на входы необходимо подать эталонное напряжение, которое можно получить, используя источники опорного напряжения ИОН или стабилитроны. Функционирование счетчиков-таймеров СТ может быть проверено, например, формированием эталонного сигнала одним из СТ и измерение его другим

Тестирование портов

Тестирование портов осуществляется последовательным их соединением с друг другом, передавая данные через один и принимая через другой.

Тестирование клавиатуры

Для проверки работоспособности клавиатуры достаточно в определенной последовательности нажимать на ее клавиши и ожидать сигнала на портах микроконтроллера. Если все значения совпали – оба устройства считаются исправными, если нет, то необходимо какое именно устройство работает некорректно. Можно взять другую клавиатуру и подключить ее к портам микроконтроллера, и нажимать в той же последовательности те же клавиши. Если после замены клавиатуры вторая клавиатура работает исправно, то это символизирует то, что первая клавиатура работала неправильно.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта был разработан алгоритм работы, структура и принципиальная схема системы, предназначенной для измерения напряжения с каналов. В процессе работы программы происходит измерение напряжения с определенного канала, который пользователь выбирает по нажатию на соответствующую клавишу на клавиатуре.

Программа была написана на языке программирования Си и отлажена в программном пакете Proteus 8 Professional. После запуска приложения просматриваем память программ и память данных в программном пакете Proteus 8 Professional. Память программ – 3,892 байт, память данных –350 байт.

Оценка времени выполнения основных алгоритмов программы дала следующие результаты:

* полный цикл опроса всех кнопок клавиатуры: t 200 мкс;
* измерение напряжения с включением светодиода: t 700 мкс;
* передача значения по USART: t 23 мкс;

Разработанная система имеет сложную и нелинейную архитектуру с программной точки зрения, чтобы в полной мере оценить и описать одной формулой время ее выполнения.

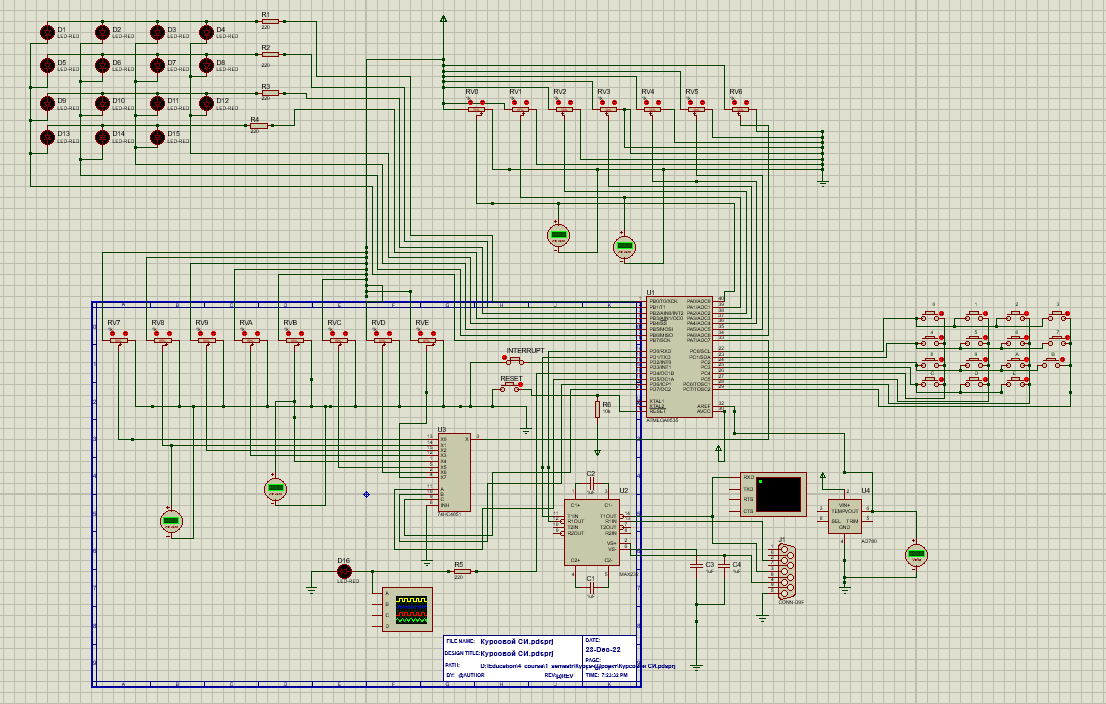
# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Иоффе, В. Г. Структурная организация однокристальных микроконтроллеров. [Электронный ресурс] : [учеб. пособие] / В. Г. Иоффе ; М-во образования и науки Рос. Федерации, Самар. нац. исслед. ун-т им. С. П. Королева (Самар. ун-т). - Самара : Изд-во Самар. ун-та, 2017. URL: <http://repo.ssau.ru/handle/Metodicheskie-materialy/Strukturnaya-organizaciya-odnokristalnyh-mikrokontrollerov-Elektronnyi-resurs-ucheb-posobie-70944> (дата обращения 20.09.22)

2. Проектирование микропроцессорных устройств на базе однокристальных микроконтроллеров [Электронный ресурс] : [метод. указания] сост. В. Г. Иоффе / М-во образования и науки Рос. Федерации, Самар. нац. исслед. ун-т им. С. П. Королева (Самар.ун-т). - Самара : Изд-во Самар. ун-та, 2015. URL:. <http://repo.ssau.ru/handle/Metodicheskie-materialy/Proektirovanie-mikroprocessornyh-ustroistv-na-baze-odnokristalnyh-mikrokontrollerov-Elektronnyi-resurs-metod-ukazaniya-70959> (дата обращения 21.10.22)

3. Разработка и отладка микропроцессорных устройств в виртуальной среде моделирования Proteus [Электронный ресурс] : [метод. указания], сост. В. Г. Иоффе / М-во образования и науки Рос. Федерации,;. Самар. нац. исслед. ун-т им. С. П. Королева (Самар. ун-т). - Самара : Изд-во Самар. ун-та, 2017. URL: <http://repo.ssau.ru/handle/Metodicheskie-materialy/Razrabotka-i-otladka-mikroprocessornyh-ustroistv-v-virtualnoi-srede-modelirovaniya-Proteus-Elektronnyi-resurs-metod-ukazaniya-70958> (дата обращения 26.10.22)

# ПРИЛОЖЕНИЕ А – Принципиальная схема



# ПРИЛОЖЕНИЕ Б – Спецификация элементов

|  |  |  |
| --- | --- | --- |
| Обозначение | Наименование | Кол-во |
|  | Резисторы |  |
| RV0..RVE | POT-HG | 15 |
| R1…R5 | CF-100, 220 Ом | 5 |
| R6 | CF-100, 10кОм | 1 |
|  | Диоды |  |
| D1…D16 | GNL-3012HD | 16 |
|  | Кнопки |  |
| 0…E, INTERRUPT, RESET |  | 17 |
|  | Микросхемы |  |
| U1 | ATMEGA 8535 | 1 |
| U2 | MAX-232 | 1 |
| U3 | 74HC4051 | 1 |
| U4 | AD780 | 1 |

# ПРИЛОЖЕНИЕ В – Листинг программы

#include <inttypes.h>

#include <avr/io.h>

#include <avr/interrupt.h>

#include <avr/sleep.h>

// Минимальные значения

short minValueChannel[15] = { 150, 150, 150, 150,

150, 150, 150, 150,

150, 150, 150, 150,

150, 150, 150};

// Максимальное значение

short maxValueChannel[15] = {180, 175, 255, 255,

255, 255, 255, 255,

255, 255, 255 160,

255, 255, 255};

// -1: Значение выходит за диапазон

// -2 : Кнопка никогда не нажималась

short valueChannel[15] = {-2,-2,-2,-2,

2,-2,-2,-2,

2,-2,-2,-2,

-2,-2,-2};

float faultADC = 4.09;

short portState[4]= {0xEF,0xDF,0xBF,0x7F};

short inputState[4]={0x01,0x02,0x04,0x08};

short cankeyChannel = 0;

short keyChannel = 0xFF;

short adcChannel = 0xFF;

short readyToGiveResult = 0;

// Опрос клавиатуры

void keyboardPolling(void)

{

short i = 0;

short j = 0;

for(i=0; i<4; i++)

{

PORTC = portState[i];

for(j=0; j<4; j++)

{

if(((PINC&inputState[j])==0))

{

while(((PINC&inputState[j])==0)) {}

if(((4 \* j) + i + 1) < 16)

{

keyChannel = (4 \* j) + i + 1;

cankeyChannel = 1;

}

}

}

}

}

// Отправка бита по USART

void sendByteByUsart(char b)

{

while(!(UCSRA &(1<<UDRE))) // Устанавливается, когда регистр свободен

{}

UDR = b;

}

// Прерывание на INT0

ISR(INT0\_vect)

{

if(readyToGiveResult == 1)

{

short i = 0;

for(i = 0; i < 15; i++)

{

short value = valueChannel[i];

if(value == -2)

{

continue;

}

sendByteByUsart((char)i);

if(value == -1)

{

sendByteByUsart(0xC0);

sendByteByUsart(0x00);

continue;

}

sendByteByUsart((char)value);

}

}

}

// Переполнение таймера

ISR(TIMER1\_COMPB\_vect)

{

PORTD ^= (1 << 4);

TCCR1B = 0;

TCNT1 = 0x0000;

}

// Опрос по клаве

ISR(TIMER0\_OVF\_vect)

{

keyboardPolling();

}

// Включение светодиода

void turnOn\_Diod(void)

{

short leftPos = (keyChannel - 1) % 4;

short rightPos = (keyChannel - 1) / 4 ;

PORTB = (0xF0 ^ (0X80 >> leftPos)) | (1 << rightPos);

}

// Выключение светоидода

void resetDiods(void)

{

PORTB = 0x00;

}

// Внешний сигнал

void startOutsideSignal()

{

PORTD ^= (1 << 4);

short i = 0;

for(i = 0; i < 1; i++);

PORTD ^= (1 << 4);

}

//ADC прерывание

ISR(ADC\_vect)

{

short adcValue = ADC / faultADC;

if(adcValue >= minValueChannel[keyChannel - 1] && adcValue <= maxValueChannel[keyChannel - 1])

{

valueChannel[keyChannel - 1] = adcValue;

resetDiods();

}

else

{

valueChannel[keyChannel - 1] = -1;

turnOn\_Diod();

}

startOutsideSignal();

readyToGiveResult = 1;

}

// Иницилазиация T0

void init\_T0(void)

{

TIMSK |= 0b00000001;

TIFR |= 0b00000001;

TCCR0 = 0b00000001;

}

// Иницилизация T1

void init\_T1(void)

{

TIMSK |= (1 << 3);

TIFR |= (1 << 3);

OCR1B =0x0028;

TCCR1B |= 0;

TCNT1 = 0;

}

// Иницилизация ADC

// 0.005

void init\_ADC(void)

{

ADCSRA = (1 << ADEN) | (1 << ADIE);

}

// Иницилизация USART

// (4000000 / (16 \* 19200)) - 1 = 12.05 = 12

void init\_UsarT(void)

{

UCSRC = (1<<URSEL)|(1<<UCSZ0)|(1<<UCSZ1);

UBRRL = 12;

UCSRB= (1<<RXEN)|(1<<TXEN)|(0<<RXCIE)|(0<<TXCIE)|(0<<UDRIE);

}

// Инилизиация портов

void init\_Ports(void)

{

DDRD |= (0 << 2);

PORTD |= (1 << 2) | (1 << 4);

GICR = (1 << INT0);

DDRA = 0x00;

DDRC = 0b11110000;

DDRB = 0b11111111;

DDRD |= (1 << 5) | (1 << 6) | (1 << 7) | (1 << 4);

}

// Передать значение с канала в ADC

void readVoltADC(void)

{

ADMUX = adcChannel - 1;

ADCSRA |= (1 << ADSC);

}

int main()

{

cli();

init\_ADC();

init\_UsarT();

init\_Ports();

init\_T0();

init\_T1();

sei();

// Write your code here

while (1)

{

if(cankeyChannel == 1)

{

cankeyChannel = 0;

adcChannel = keyChannel;

if(keyChannel > 7)

{

PORTD &= 0X1F;

PORTD |= (adcChannel - 8) << 5;

adcChannel = 8;

}

readVoltADC();

}

}

return 0;

}