

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МЭИ»
в г. Смоленске

Кафедра
электроники и микропроцессорной техники

КУРСОВОЙ ПРОЕКТ

по дисциплине «Основы микропроцессорной техники»
Тема: **РАЗРАБОТКА АППАРАТНЫХ И ПРОГРАММНЫХ СРЕДСТВ
ЭЛЕКТРОННЫХ УСТРОЙСТВ НА ОСНОВЕ
МИКРОКОНТРОЛЛЕРОВ**

Студент группы ПЭ1-16

дата сдачи

подпись

Петров П.П.

Руководитель

к.т.н., доцент
Амелина М.А.

подпись

Проект допущен к защите

к.т.н., доцент
Амелина М.А.

дата

подпись

Дата защиты

Оценка

Подписи

членов комиссии

Смоленск 2019

АННОТАЦИЯ

Автор работы: Петров Павел Петрович.

Тема — разработка измерителя концентрации паров спирта в промилле на литр крови по содержанию паров этанола в выдыхаемом воздухе.

В проекте использовался газоанализаторный модуль на основе датчика MQ-3 и устройство индикации на основе четырехразрядного семисегментного дисплея с общими катодами. В качестве управляющего процессом измерения и динамической индикацией устройства использован микроконтроллер ATmega16. Реализация динамической индикации программно-аппаратная на основе четырех катодных ключей и периодического таймерного прерывания, изменяющего номер индицируемого знакоместа. Предусмотрена звуковая сигнализация при превышении допустимого порога (0,3 промилле).

В курсовом проекте разработана функциональная и принципиальная схемы измерителя, алгоритм работы устройства, выполнен расчёт параметров устройства динамической индикации, приведена реализация алгоритма в виде программы на языке СИ.

Расчетно-пояснительная записка содержит 43 страницы, 28 рисунков, 1 таблицу, 2 приложения.

Применено программное обеспечение Microsoft Word 2013, Altium Designer 14.3, CodeVisionAVR 2.13, Proteus 7.10.

СОДЕРЖАНИЕ

<u>ТЕХНИЧЕСКОЕ ЗАДАНИЕ</u>	4
<u>ВВЕДЕНИЕ</u>	5
<u>1 РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА</u>	7
<u>1.1 Анализ технического задания</u>	7
<u>1.2 Структурная схема</u>	8
<u>1.3 Функциональная схема</u>	9
<u>2 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ СХЕМЫ</u>	14
<u>2.1 Микроконтроллер</u>	14
<u>2.1.1 Схема тактирования</u>	14
<u>2.1.2 Схема сброса микроконтроллера</u>	15
<u>2.2 Датчик газа</u>	18
<u>2.3 Схема динамической индикации</u>	19
<u>2.3.1 Расчет токоограничивающих резисторов светодиодных сегментов</u>	19
<u>2.3.2 Расчет базовых резисторов катодных ключей</u>	19
<u>2.4 Звуковая сигнализация</u>	20
<u>2.5 Обеспечение помехоустойчивости устройства</u>	20
<u>3 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ</u>	23
<u>3.1 Измерение концентрации газовой примеси с помощью газоанализаторных датчиков серии MQ</u>	23
<u>3.2 Управление динамической индикацией</u>	27
<u>3.3 Блок-схема алгоритма</u>	28
<u>3.4 Программа на языке СИ</u>	30
<u>4 ПРОВЕРКА РАБОТОСПОСОБНОСТИ УСТРОЙСТВА</u>	33
<u>4.1 Моделирование в среде PROTEUS</u>	33
<u>4.2 Проверка работы измерителя концентрации спирта с помощью отладочной платы</u>	36
<u>ЗАКЛЮЧЕНИЕ</u>	38
<u>СПИСОК ИСТОЧНИКОВ</u>	39
<u>ПРИЛОЖЕНИЕ А СХЕМА ЭЛЕКТРИЧЕСКАЯ ПРИНЦИПИАЛЬНАЯ</u>	40
<u>ПРИЛОЖЕНИЕ Б ПЕРЕЧЕНЬ ЭЛЕМЕНТОВ</u>	42

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Разработать измеритель концентрации паров спирта на основе газоанализаторного датчика MQ-3 с выводом информации на монолитный четырехразрядный семисегментный индикатор с общими катодами. Динамическую индикацию организовать программно. Предусмотреть звуковую сигнализацию при превышении допустимого порога.

Состав задания на курсовой проект:

- Разработать принципиальную схему устройства;
- Разработать программу для микроконтроллера, реализующую требования технического задания;
- Произвести отладку разработанной программы;
- Промоделировать работу устройства в программе PROTEUS;
- Проверить работу устройства с помощью отладочной платы на основе МК ATmega16;
- Создать пакет конструкторской документации на разработанное устройство, включающий принципиальную схему и перечень элементов.

ВВЕДЕНИЕ

Проектирование современных электронных устройств невозможно представить без использования микроконтроллеров. Раньше инженеру для создания даже тривиального устройства необходимо было создавать целый комплекс из микросхем средней степени интеграции, выполняющих вместе заданную функцию. При этом необходимо было, чтобы все микросхемы были бы однотипной элементной базы, которую также необходимо было выбрать.

Сейчас, благодаря тому, что арсенал проектировщика пополнился таким устройством как микроконтроллер (МК), ему достаточно, в большинстве случаев разработать алгоритм работы устройства, после этого добавить на схему необходимый «обвес» для него (индикаторы, датчики, разъемы для связи по определенному протоколу и т.п.), и наконец, на последнем этапе правильно запрограммировать МК.

Микроконтроллеров существует очень много. А для разработки определённого устройства требуется выбрать из этого огромного множества что-то конкретное. Для этого существует строгая система. МК делятся на семейства, что существенно облегчает процесс выбора, ведь у каждого семейства есть свои достоинства и недостатки.

Для выполнения задания была выбран МК ATMEGA16 семейства AVR от компании Atmel. Контроллеры этого семейства доступны и в России и за рубежом, не очень дорого стоят, к тому же имеют разнообразную периферию (таймеры, АЦП, различные последовательные интерфейсы и т.д.), обширную систему команд (около 130) и, что немаловажно, имеются в наличии бесплатные компиляторы СИ (WINAVR, AtmelStudio, CodeVisionAVR demo, Arduino IDE). Для реализации технического задания был выбран компилятор CodeVisionAVR V3.34 Evaluation — простой и удобный компилятор, имеющий функцию автоматической предварительной генерации кода [1, 2, 3, 5]. Кроме того, данный

компилятор изучался в течение семестра в курсе лабораторных работ [1, 5] и практических занятий, так что работа с ним и его основные особенности уже известны.

После написания программы необходимо проверить, работоспособна ли она. Для этого удобнее всего использовать программу схемотехнического моделирования Proteus VSM, разработанную компанией Labcenter Electronics. Широко используются 2 версии программы PROTEUS. Более старая версия PROTEUS 7.10 проста в использовании, надежна в работе, содержит минимальный набор моделей датчиков, которые могут подключаться к микроконтроллеру. Поддерживает создание проекта лишь на языке низкого уровня (асемблере) внутри модельного файла. Новая версия PROTEUS 8.7 содержит более широкий набор моделей внешних устройств, подключаемых к МК; кроме того, поддерживает для МК AVR программирование на языке Arduino IDE.

Для проверки работоспособности в «железе» будет использоваться отладочная плата на основе МК Atmega16 (рис. В.1). К ней через параллельные порты и аналоговый вход будет подключаться необходимая периферия: семисегментный индикатор для мультиплексного режима, звуковой сигнализатор и газоанализаторный датчик.

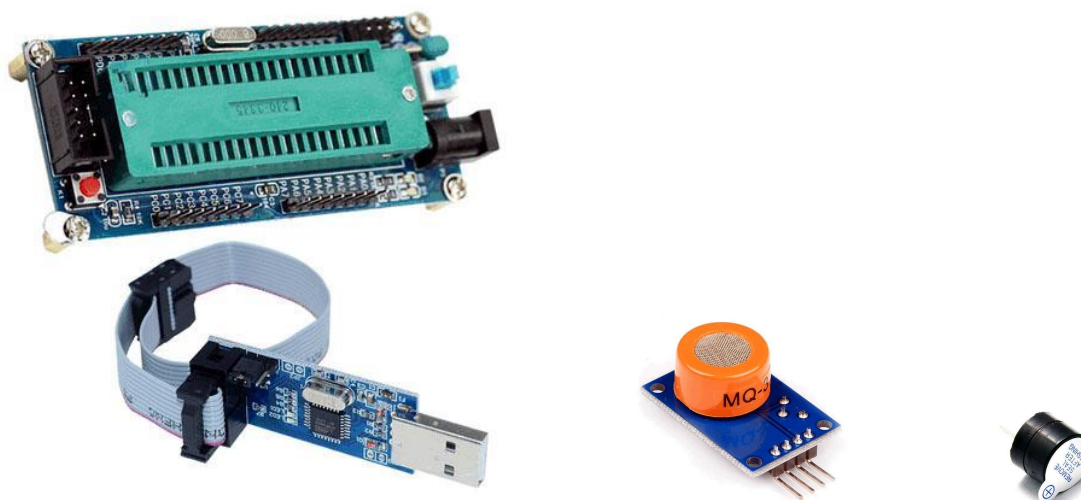


Рисунок В.1 Отладочная плата для МК АТmega16, программатор USBASP, датчик MQ-3 и зуммер

Также может использоваться для создания тестового макета и отладочная плата EasyAVR5A [1, 4]. На ней уже есть семисегментный 4хразрядный индикатор с катодными ключами, а газоанализаторный датчик и зуммер можно подключить к расположенным на плате штыревым разъемам параллельных портов.

1 РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА

1.1 Анализ технического задания

Согласно ТЗ стоит задача измерения паров этанола в воздухе. Для измерения концентрации будем использовать соответствующий датчик MQ-3. MQ-3 относится к полупроводниковым приборам. Принцип работы датчика основан на изменении сопротивления тонкопленочного слоя диоксида олова SnO_2 при контакте с молекулами определяемого газа. Чувствительный элемент датчика состоит из керамической трубки с покрытием Al_2O_3 и нанесенного на неё чувствительного слоя диоксида олова. Внутри трубки проходит нагревательный элемент, который нагревает чувствительный слой до температуры, при которой он начинает реагировать на определяемый газ (рис. 1.1).

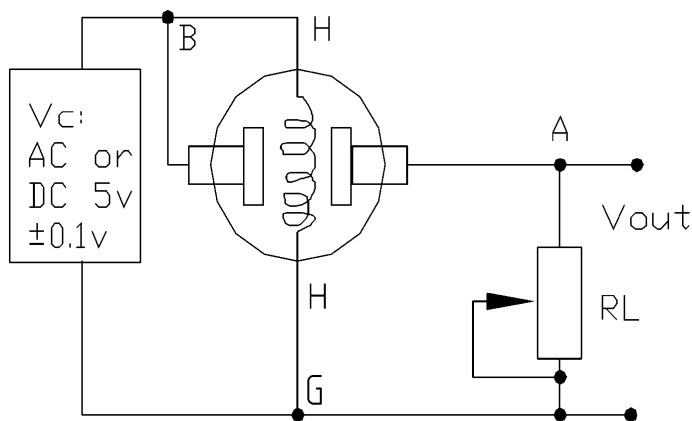


Рисунок 1.1 Структура датчика MQ-3

В датчике MQ-3 будут использоваться 3 внешних вывода:

- Сигнальный (S) — выходной сигнал сенсора (A, Vout на рис. 1.1). Подключается к аналоговому входу микроконтроллера (входу АЦП).
- Питание (B на рис. 1.1) — питание датчика. Соединяется с положительным выводом источника питания микроконтроллера.
- Земля (G) — соединяется с выводом GND микроконтроллера.

При увеличении концентрации примесного газа сопротивление сенсора R_s уменьшается. Поскольку сопротивление сенсора включено в верхнее плечо делителя напряжения (рис. 1.1), а на делитель (как и на нагревательный элемент) подано напряжение питания, то напряжение на нижнем плече делителя V_{OUT} будет увеличиваться.

Это напряжение может быть преобразовано в цифровую форму с помощью АЦП микроконтроллера ATmega16, затем программно обработано для получения концентрации примеси в соответствующих единицах (либо в мг/литр воздуха, либо в промилле на литр крови). Заключительным этапом будет являться вывод концентрации на индикацию.

В качестве индикатора задан четырехразрядный монолитный семисегментный индикатор с общими катодами, что предполагает организацию динамической (мультиплексной) индикации. Следовательно, дополнительно необходимо будет использовать транзисторные катодные ключи для выбора знакоместа, а также периодически программно организовывать смену индицируемых знакомест. Смену индицируемого знакоместа через строго определённый интервал времени удобнее организовать с помощью прерывания по совпадению таймера, присутствующего в МК ATmega16.

Разработка устройства производится в несколько этапов:

- разработка структурной и функциональной схем устройства;
- выбор или разработка алгоритма работы устройства;
- разработка принципиальной схемы устройства;
- написание программы для микроконтроллера и её отладка;
- проверка работоспособности устройства (путем моделирования в среде PROTEUS и тестирования на отладочной плате);
- конструкторская проработка устройства.

1.2 Структурная схема

Исходя из сказанного выше (см. подраздел 1.1) структурная схема будет иметь вид (рис. 1.2).

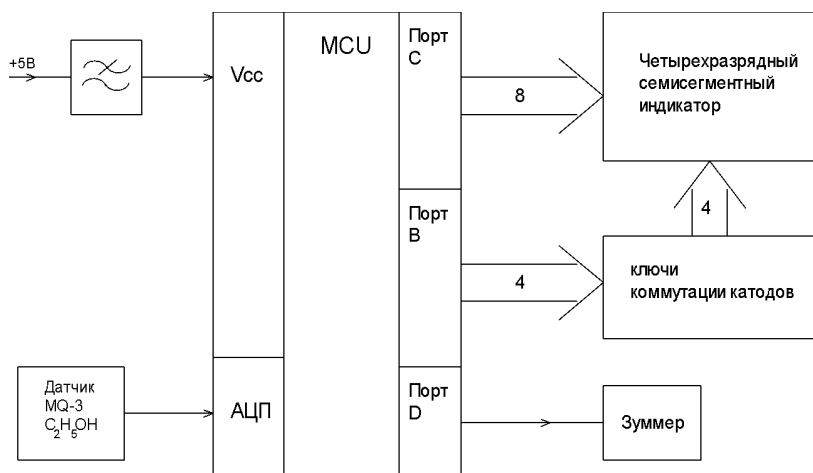


Рисунок 1.2 Структурная схема измерителя паров газовой примеси

Понадобится микроконтроллер, имеющий в своём составе следующую периферию:

- 8- или 16-разрядный таймер. Необходим для организации периодической смены знакомест с частотой не меньшей $4 \cdot f_{\text{КЧМ}}$, т.е. через интервал времени $T_{\text{ЗН}} \leq 1/(4 \cdot f_{\text{КЧМ}})$.
- Аналого-цифровой преобразователь разрядностью не меньшей 8 бит.
- Порты параллельного ввода-вывода с разрядностью 13 (8 для управления анодами сегментов семисегментного индикатора, 4 — для выбора индицируемого знакоместа с помощью катодных ключей, 1 — для подачи звуковой сигнализации).

Для питания составных частей устройства необходим источник питания или внешнее кондиционированное напряжение питания. Напряжением 5 В будет запитываться микроконтроллер и датчик газовой примеси.

1.3 Функциональная схема

На данном этапе желательно выбрать микроконтроллер для точного уяснения функциональных связей между составными частями устройства. Необходимым требованиям (см. подраздел 1.2) удовлетворяет микроконтроллер Atmega16 [6], имеющий Гарвардскую архитектуру и обширный набор встроенной периферии (рис. 1.3).

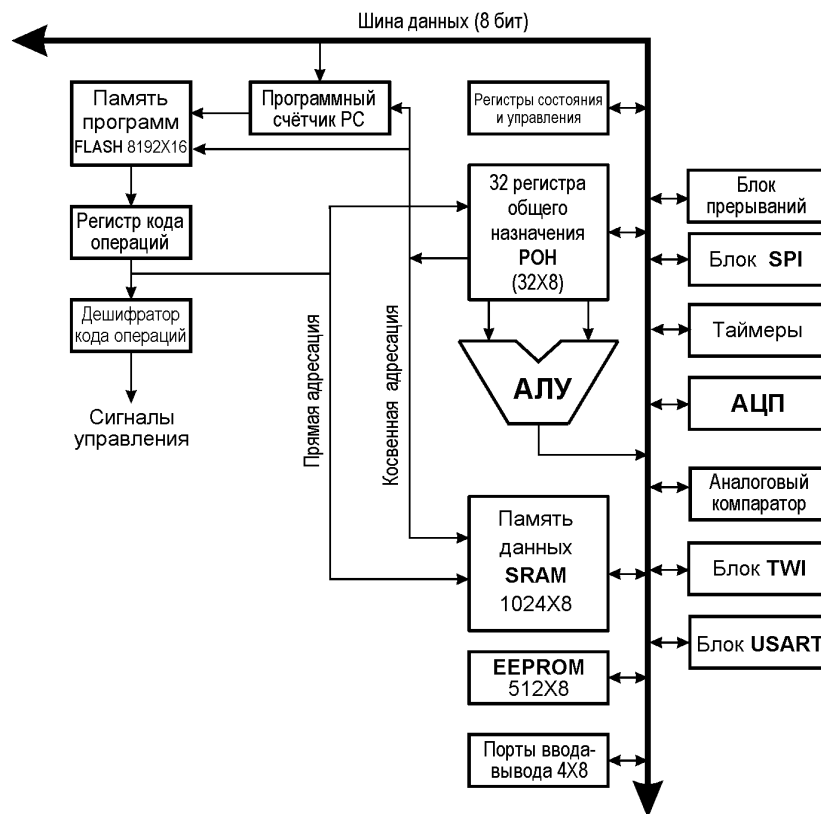


Рисунок 1.3 Внутренняя структура микроконтроллера ATmega16

Данный микроконтроллер имеет в своём составе 2 8-разрядных и 1 16разрядный таймеры, могущие работать в 4-х различных режимах. Для решаемой в проекте задачи — периодической смены индицируемых на семисегментном индикаторе знакомест можно использовать 8-разрядный таймер в режиме CTC (Clear to Compare Match — сброс при совпадении), организовав соответствующее периодическое прерывание по совпадению через необходимый временной интервал. В обработке прерывания будет организована циклическая смена включаемых катодных ключей с помощью

соответствующих линий параллельного порта ввода-вывода. В результате с большой скоростью (незаметной для глаза человека) выводимые цифры десятичного числа на знакоместах индикатора будут индицироваться по очереди, а наблюдатель будет воспринимать изображение как непрерывное, состоящее из 4 десятичных цифр.

Микроконтроллер Atmega16 имеет в своем составе 10разрядный АЦП последовательного приближения, что также делает его подходящим для решения поставленной задачи, поскольку с датчика поступает аналоговое напряжение, зависящее от концентрации примесного газа. АЦП будет осуществлять преобразование этого напряжения в цифровой код, далее по нему будет рассчитываться искомая концентрация газовой примеси и выводиться на индикацию.

Рассматриваемый микроконтроллер имеет 4 8-разрядных порта параллельного ввода-вывода, чего с избытком достаточно для параллельного ввода-вывода, используемого в проекте — 13 линий.

Кроме того, немаловажным фактором, повлиявшим на выбор МК, является то, что для тестирования разработанных устройств имеется в распоряжении отладочные платы: EasyAVR5A (рис. 1.4) и плата для МК Atmega16 (см. рис. В.1), укомплектованные микроконтроллером указанного типа.

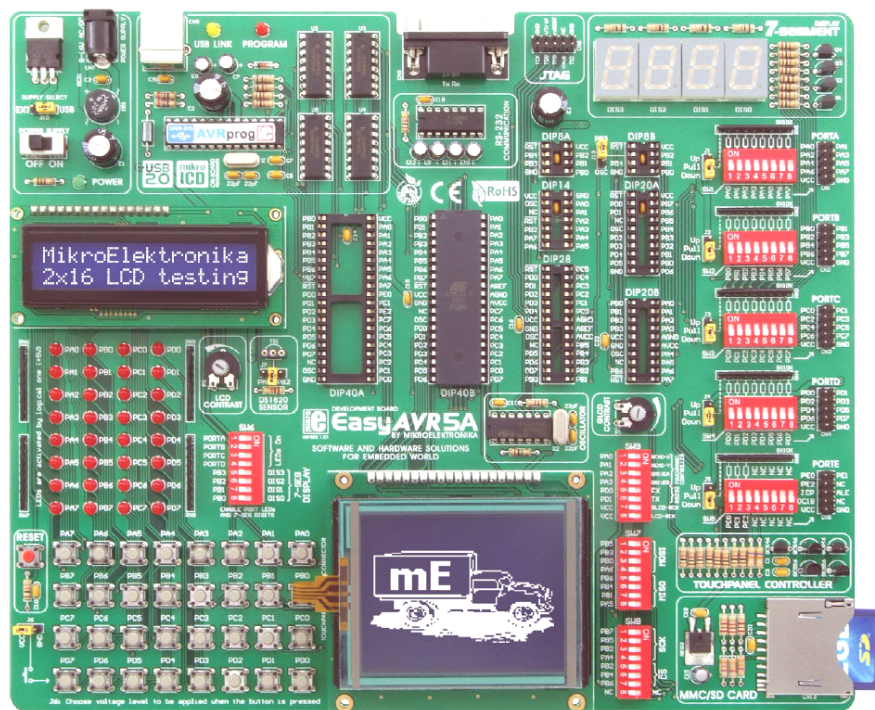


Рисунок 1.4 Отладочная плата EasyAVR5A

Микроконтроллер работает по программе, реализующей алгоритм измерения и преобразования данных и их динамической индикации на 4-разрядном семисегментном индикаторе. Программа записана во Flash-памяти программ, для временного хранения промежуточных переменных используется регистровая и оперативная память МК (рис. 1.3).

Главный модуль программы (а вернее его бесконечный цикл — фоновая программа) осуществляет периодическое преобразование аналогового напряжения, поступающего на вход АЦП в цифровой код и расчет концентрации газовой примеси. Также для полученного значения концентрации осуществляется преобразование в массив десятичных цифр для отображения их на выводном устройстве — семисегментном индикаторе.

Второй модуль программы — обработчик прерывания по совпадению отвечает за вывод этого массива десятичных цифр на семисегментный индикатор в режиме динамической (мультиплексной) индикации. При этом младшая тетрада порта В используется для управления катодными ключами, а

порт С — для вывода графики знака (зажигания необходимых сегментов) на выбранном с помощью катодного ключа знакоместе.

Исходя из сказанного функциональная схема устройства будет иметь вид рис. 1.5.

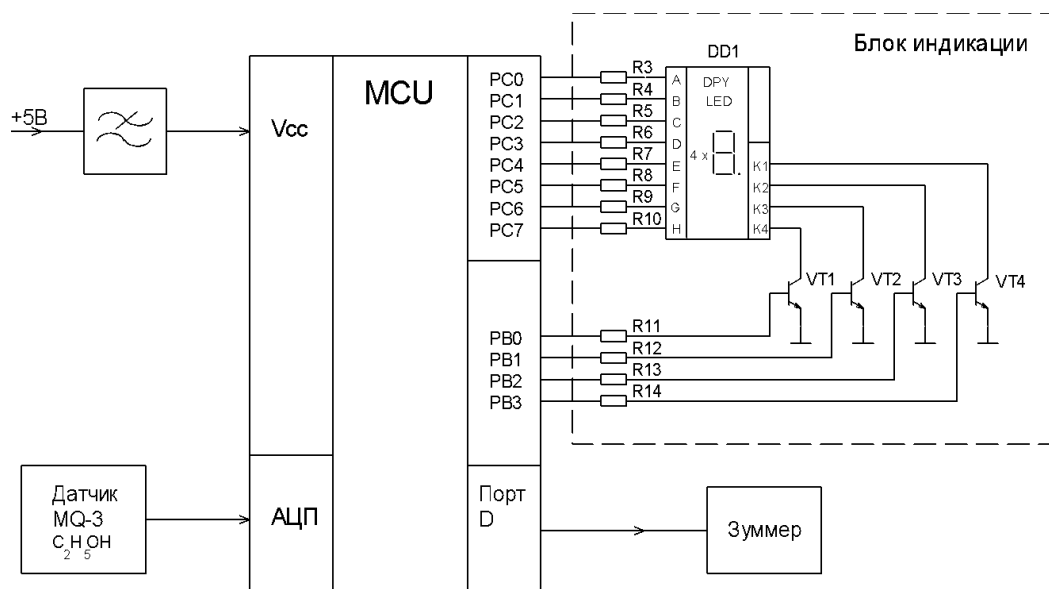


Рисунок 1.5 Функциональная схема измерителя концентрации этанола

Устройство работает следующим образом. Вначале (перед заходом в вечный цикл фоновой программы) происходит 5-минутный прогрев датчика и его калибровка при существующих условиях (температуре, давлении, влажности) в чистом воздухе. После вхождения в вечный цикл происходит циклическое аналого-цифровое преобразование напряжения с датчика, его обработка и вычисление с помощью аппроксимирующей формулы концентрации газовой примеси. При превышении допустимой концентрации (0,25 промилле/литр) выдается звуковой сигнал. Также в вечном цикле происходит преобразование полученного значения концентрации в массив десятичных цифр размерностью 4X1 байт для подготовки к выводу данных на семисегментный индикатор в мультиплексном режиме.

Таймер 0 программируется в режиме CTC с разрешением прерывания по совпадению. В результате через заданное время (определяемое частотой тактирования таймера и значением в регистре сравнения OCR0) будут

возникать внутренние запросы соответствующего прерывания от таймера. Обработчик этого прерывания фактически и организует динамическую индикацию измеренного значения. Он осуществляет выбор текущего индицируемого знакоместа индикатора путем включения необходимого катодного ключа подачей логической единицы на соответствующий вывод младшей тетрады порта В. Далее через порт С выводится код графики (зажигаются необходимые сегменты) цифры из массива цифр для измеренной концентрации. После этого данный модуль заканчивает свою работу — происходит возврат в фоновую программу.

Работа по программе процессора происходит под управлением внутренних часов МК – тактового генератора, от него же берутся импульсы для тактирования таймера T0. Импульсы тактового генератора микроконтроллера (частотой 12 МГц) делятся предделителем таймера на 256, таким образом получаются его тактовые импульсы частотой 46,875 кГц. Соответствующей инициализацией таймера T0 добиваются того, что прерывания по совпадению от таймера будут возникать через заданное время — время индикации одного знакоместа при развертке всего изображения из четырех десятичных цифр $T_{\text{зн}} \leq 0,25 \cdot (1/f_{\text{кчм}})$.

Питается микроконтроллер и датчик от внешнего питающего напряжения +5 В, подаваемого через разъём. От него же берётся питание для блока аналогового компаратора и АЦП микроконтроллера и опорное напряжение АЦП.

2 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ СХЕМЫ

2.1 Микроконтроллер

Микроконтроллер выполняет программу, написанную на языке СИ, реализующую алгоритм циклического измерения концентрации газовой примеси в окружающем воздухе. В проекте используется МК Atmega16, внутренняя структура которого представлена на рис. 1.3.

2.1.1 Схема тактирования

Для того, чтобы микроконтроллер мог работать по программе, записанной в его Flash-память программ, его необходимо тактировать. В проекте используется внешнее тактирование кварцевым резонатором внутреннего тактового генератора.

Резонатор подключается к выводам XTAL1 и XTAL2 микроконтроллера, как показано на рис. 2.1. Эти выводы являются соответственно входом и выходом инвертирующего усилителя тактового генератора. Емкости конденсаторов C2 и C3, подключаемых между выводами резонатора и общим проводом, зависят от типа резонатора. Для кварцевых резонаторов емкости этих конденсаторов находятся в пределах 12–22 пФ, а для керамических резонаторов должны выбираться согласно рекомендациям производителей резонаторов.

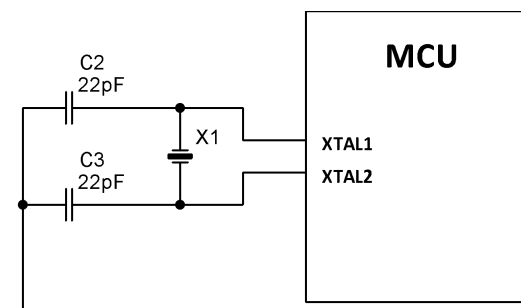


Рисунок 2.1 Схема тактирования микроконтроллера

Усилитель тактового генератора МК ATmega16 может работать в одном из двух режимов, определяемом состоянием конфигурационной ячейки СКРОТ.

Если эта ячейка запрограммирована («0»), размах колебаний на выходе усилителя (вывод XTAL2) практически равен напряжению питания.

Данный режим полезен при работе устройства в условиях сильных электромагнитных помех, а также при использовании сигнала тактового генератора для управления внешними устройствами. В последнем случае между выводом и внешней схемой обязательно должен быть буфер.

Если ячейка СКРОТ не запрограммирована («1»), размах колебаний на выходе усилителя будет значительно меньше. Соответственно ток потребления микроконтроллера уменьшается, однако при этом сужается и диапазон возможных частот тактового сигнала. Кроме того, в этом режиме сигнал тактового генератора микроконтроллера нельзя использовать для управления внешними устройствами.

Собственно генератор может работать в четырех различных режимах [6], каждый из которых предназначен для определенного диапазона частот. Эти режимы определяются ячейками CKSEL3...1 и СКРОТ (табл. 2.1).

Таблица 2.1 Режимы работы кварцевого генератора

СКРОТ	CKSEL3...1	Примерный диапазон частот [МГц]
1	101*	0.4...0.9
1	110*	0.9...3.0
1	111	3.0...12.0
0	101, 110, 111	> 1.0
* В этом режиме должен использоваться только керамический резонатор.		

В соответствии с типовой частотой работы МК Atmega16 и с тем, что сигнал тактовой частоты не будет использоваться для управления внешними устройствами, выбираем режим, соответствующей 3-ей строке табл. 2.1.

2.1.2 Схема сброса микроконтроллера

В состав всех микроконтроллеров семейства Mega входит система сброса по включению питания (схема POR, Power-on Reset). Эта схема удерживает микроконтроллер в состоянии сброса до тех пор, пока напряжение питания не

превысит некоторого порогового значения V_{POT} . При достижении напряжением питания значения V_{POT} схема POR запускает таймер задержки сброса. По окончании счета (после формирования задержки t_{TOUT}) внутренний сигнал сброса снимается и происходит запуск микроконтроллера.

Управлением выводом $\overline{RESET}$ микроконтроллера при включении питания может осуществляться двумя способами [6]. Если время нарастания напряжения источника питания известно и не превышает величины t_{TOUT} , можно использовать первый способ, при котором напряжение на выводе $\overline{RESET}$ «повторяет» напряжение питания (рис. 2.2). В этом случае вывод $\overline{RESET}$ можно подключить к источнику питания либо оставить неподключенным, поскольку он уже подсоединен к источнику питания подтягивающим резистором.

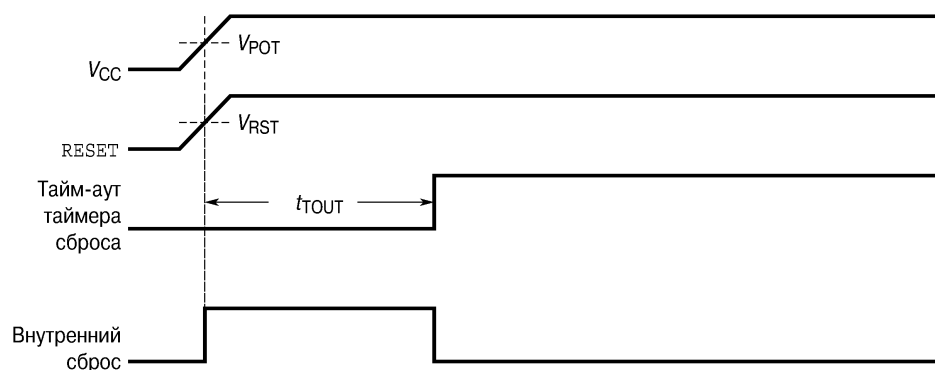


Рисунок 2.2 Временные диаграммы сигналов при сбросе в момент включения питания; вывод $\overline{RESET}$ подключен к V_{DD}

При втором способе управление выводом $\overline{RESET}$ осуществляется внешней схемой, и сигнал ВЫСОКОГО уровня подается на него только после установления напряжения питания (рис. 2.3). В этом случае работой таймера задержки сброса будет управлять схема аппаратного сброса, и он начнет работать при достижении напряжения на выводе $\overline{RESET}$ порогового значения V_{RST} .

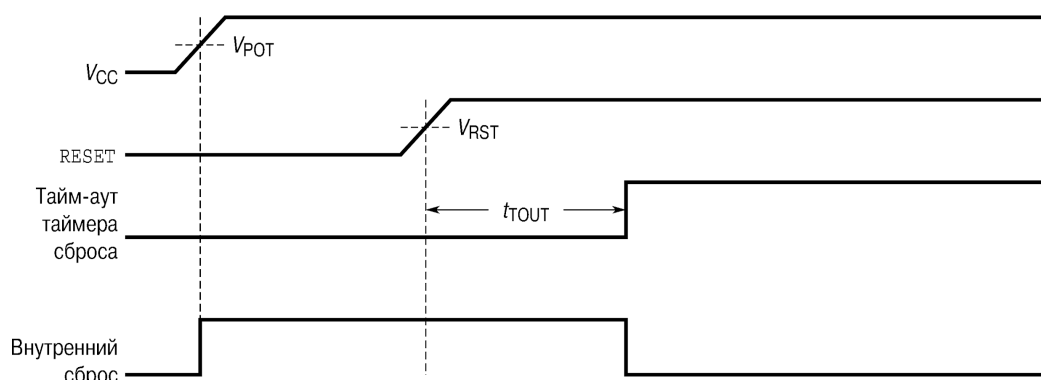


Рисунок 2.3 Временные диаграммы сигналов при сбросе по включению питания; вывод RESET управляется внешней схемой

Данное решение является более дорогостоящим, т. к. требует применения внешних компонентов. Однако этот способ позволяет «подгонять» время запуска микроконтроллера под время нарастания напряжения используемого источника питания.

Аппаратный (или внешний) сброс микроконтроллера осуществляется подачей на вывод $\overline{\text{RESET}}$ сигнала НИЗКОГО уровня. Микроконтроллер остается в состоянии сброса до тех пор, пока на выводе $\overline{\text{RESET}}$ будет присутствовать сигнал НИЗКОГО уровня. Длительность импульса сброса должна быть не менее 500 нс, в противном случае сброс микроконтроллера не гарантируется. При достижении напряжением на выводе $\overline{\text{RESET}}$ порогового значения V_{RST} запускается таймер задержки сброса. После формирования задержки t_{TOUT} внутренний сигнал сброса снимается и происходит запуск микроконтроллера (рис. 2.4).

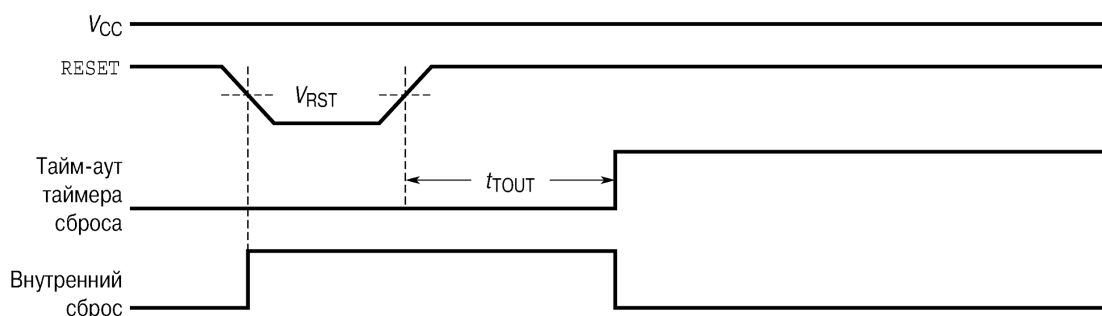


Рисунок 2.4 Временные диаграммы сигналов при аппаратном сбросе

Выбираем 3-ий вариант аппаратного сброса микроконтроллера (рис. 2.5), предоставляющий возможность нажатием кнопки сбросить микроконтроллер и заставить его перейти к началу программы (вектору сброса). Схема работает следующим образом. При включении устройства (или после кратковременного замыкания кнопки S1) конденсатор C1 разряжен и напряжение на выводе $\overline{\text{RESET}}$ близко к нулю — микроконтроллер не стартует, т.к. на него подается сигнал непрерывного сброса. Но со временем конденсатор C1 через резистор R2 зарядится и напряжение на $\overline{\text{RESET}}$ достигнет лог. «1» — МК запустится. Кнопка путем кратковременного замыкания позволяет принудительно сделать сброс, если необходимо. Резистор R1 ограничивает ток разряда C1.

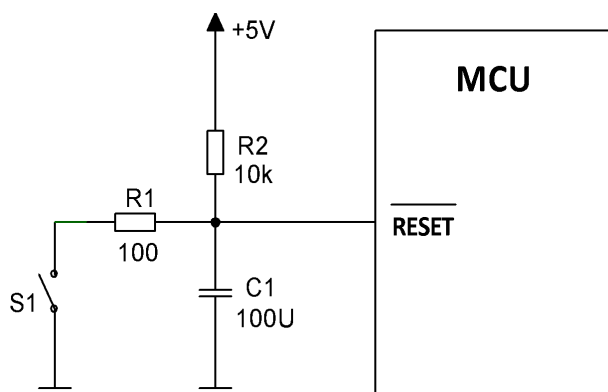


Рисунок 2.5 Схема сброса микроконтроллера

Задержка старта при сбросе (или при подаче питания) будет $T = R2 \cdot C1$, для данного примера — около секунды. Задержка нужна для того, чтобы МК не стартовал раньше, чем все устройства микроконтроллерной системы запитаются и выйдут на установившийся режим работы.

2.2 Датчик газа

Нагреватель датчика (выводы H) подключается к выводам напряжения питания +5V и GND (рис. 2.6, а). Этим обеспечивается постоянный прогрев датчика до температуры, при которой он может реагировать на наличие определённых газовых примесей (в нашем случае этанола). Один из выводов сенсора (элемента, меняющего своё сопротивление при наличии газовых

примесей) подключается к плюсовому выводу внешнего питания, второй, соединенный с постоянным резистором нижнего плеча делителя — ко входу АЦП канала 0 (PA0) микроконтроллера. В результате образуется резистивный делитель с переменным сопротивлением верхнего плеча (сопротивлением сенсора R_s), схема которого представлена на рис. 2.6, б.

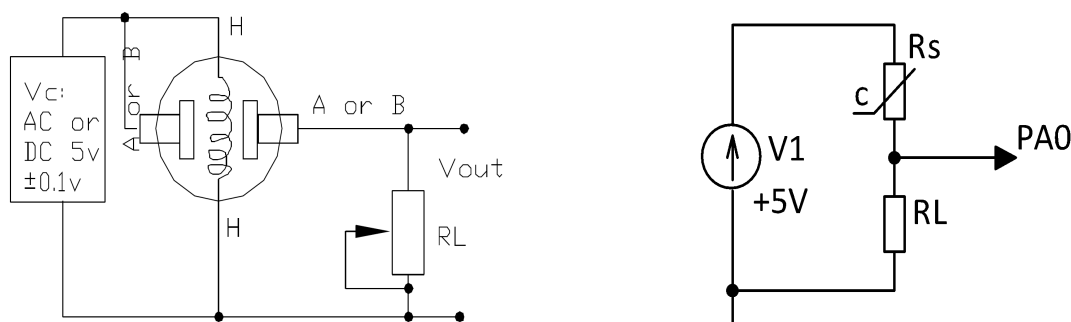


Рисунок 2.6 Схема газоанализаторных датчиков серии MQ

2.3 Схема динамической индикации

Для последующего написания программного обеспечения примем следующее. Катод старшего знакоместа семисегментного индикатора (крайнего слева) управляется линией PB3 порта В, катод младшего знакоместа (крайнего справа) — линией PB0 порта В. Соответственно катоды 2-го и 3-го знакомест, считая слева управляются линиями PB2, PB1 порта В.

Анодные сегменты (А, В, С, D, Е, F, G, DP) подсоединяются соответственно к линиям PC0...PC7. С учётом этого принципиальная схема индикации будет иметь вид (рис. 2.8).

2.3.1 Расчет токоограничивающих резисторов светодиодных сегментов

Токоограничивающие резисторы R2–R9 и светодиодные сегменты в результате организации динамической индикации по 4-м знакоместам работают в импульсном режиме, с коэффициентом заполнения $D=0,25$. Это означает, что средний ток подсветки светодиодного сегмента (от которого будет зависеть яркость) будет определяться по формуле:

$$I_{LEDavg} = I_{LEDmax} \cdot D \quad \text{или} \quad I_{LEDmax} = \frac{I_{LEDavg}}{D} \quad (2.1)$$

Задаваясь средним током сегмента $I_{LEDavg}=2$ мА, из формулы (2.1) можно определить $I_{LEDmax}=8$ мА.

Расчет резисторов R2–R9 выполняется по формуле:

$$R2 = \frac{V^1 - V_{LED} - V_{VTocm}}{I_{LEDmax}} = \frac{4,5 - 2 - 0,3}{0,008} = 275 \quad \text{Ом.}$$

Выбираем из стандартного ряда значение для R2–R7 270 Ом.

2.3.2 Расчет базовых резисторов катодных ключей

Расчёт резистров в цепи базы катодных ключей VT1–VT4 проводится из следующих соображений. Ток насыщения ключей был задан ранее и равен $I_{VTнас}=I_{LEDmax}$. Зададимся усредненным значением коэффициента передачи тока базы в маломощном транзисторе $\beta=100$. Расчёт резисторов в VT1–VT4 будем вести исходя из заданной степени насыщения $S=3$ транзисторного ключа:

$$I_{Bнас} = \frac{V^1 - V_{БЭ}}{R10} = \frac{I_{VTнас}}{\beta} S$$

Отсюда:

$$R10 = \frac{(V^1 - V_{БЭ}) \cdot \beta}{I_{VTнас} \cdot S} = \frac{(4,5 - 0,7) \cdot 100}{0,008 \cdot 3} = 16 \text{ кОм}$$

Выбираем сопротивления резисторов R10–R13 с запасом в меньшую сторону из стандартного ряда E12 R10=12 кОм.

2.4 Звуковая сигнализация

Зуммер, подключается к выводу параллельного порта PD0 (рис. 2.8) и издает сигнал звуковой частоты при подаче на него напряжения логической единицы.

2.5 Обеспечение помехоустойчивости устройства

В цифровых схемах при переключениях часто изменяются состояния логических элементов и возникают значительные переходные токи. Это приводит к кратковременным провалам в напряжении питания, а также — к излучению радиочастотной энергии в широком диапазоне частот. При больших перепадах напряжения питания может возникнуть ситуация, когда уровни цифровых сигналов примут неопределённое состояние, в результате чего конечные автоматы цифровых устройств будут работать с ошибками.

В этом случае необходимы дополнительные источники заряда (конденсаторы), позволяющие уменьшить последствия переходных процессов, стабилизируя напряжение источника питания.

Для этого в цепь питания каждой микросхемы включается безындукционный развязывающий (блокирующий) конденсатор, а между шинами питания — общий конденсатор большей емкости для подавления низкочастотных помех.

Обычно для подавления низкочастотных помех бывает достаточно применить один алюминиевый или танталовый электролитический конденсатор у входного разъема питания. Обычно его ёмкость составляет около 10 мкФ. Дополнительный керамический конденсатор будет подавлять высокочастотные помехи от других плат.

Конденсаторы, подавляющие высокочастотные помехи, должны устанавливаться как можно ближе к выводам питания микросхемы. Как правило, ёмкость такого конденсатора выбирается равной 0,1 мкФ, хотя иногда используют и конденсаторы ёмкостью 1 мкФ (в низкочастотных схемах) или 0,01 мкФ (в высокочастотных). Все конденсаторы должны иметь по возможности минимальную длину выводов. Более короткие выводы имеют меньшую индуктивность, что снижает вероятность возникновения резонанса в

схеме. Конденсатор меньшей ёмкости должен располагаться как можно ближе к микросхеме, а конденсатор большей ёмкости — как можно ближе к источнику питания (рис. 2.7).

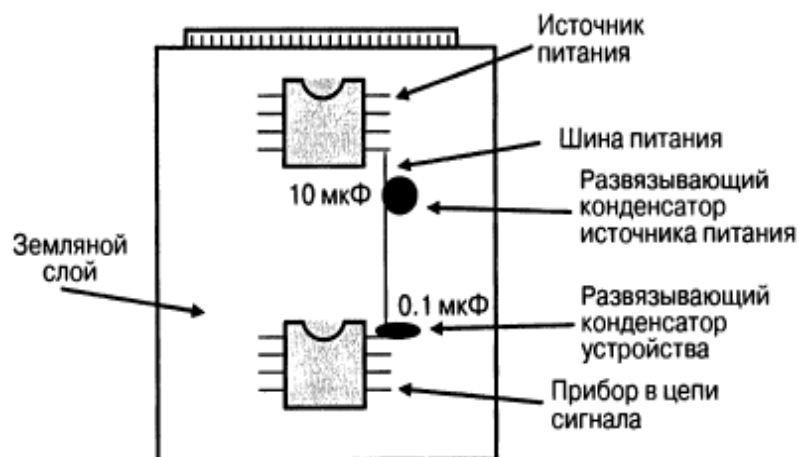


Рисунок 2.7 Расположение развязывающих конденсаторов на печатной плате

С учетом сказанного в разработанное устройство следует установить на выводы подачи питающих напряжений микроконтроллера конденсаторы емкостью 0,1 мкФ (по одному на каждый вывод).

В разработанной принципиальной схеме (рис. 2.8) учтены все рассмотренные выше связи между компонентами и предусмотрены меры по обеспечению помехоустойчивости.

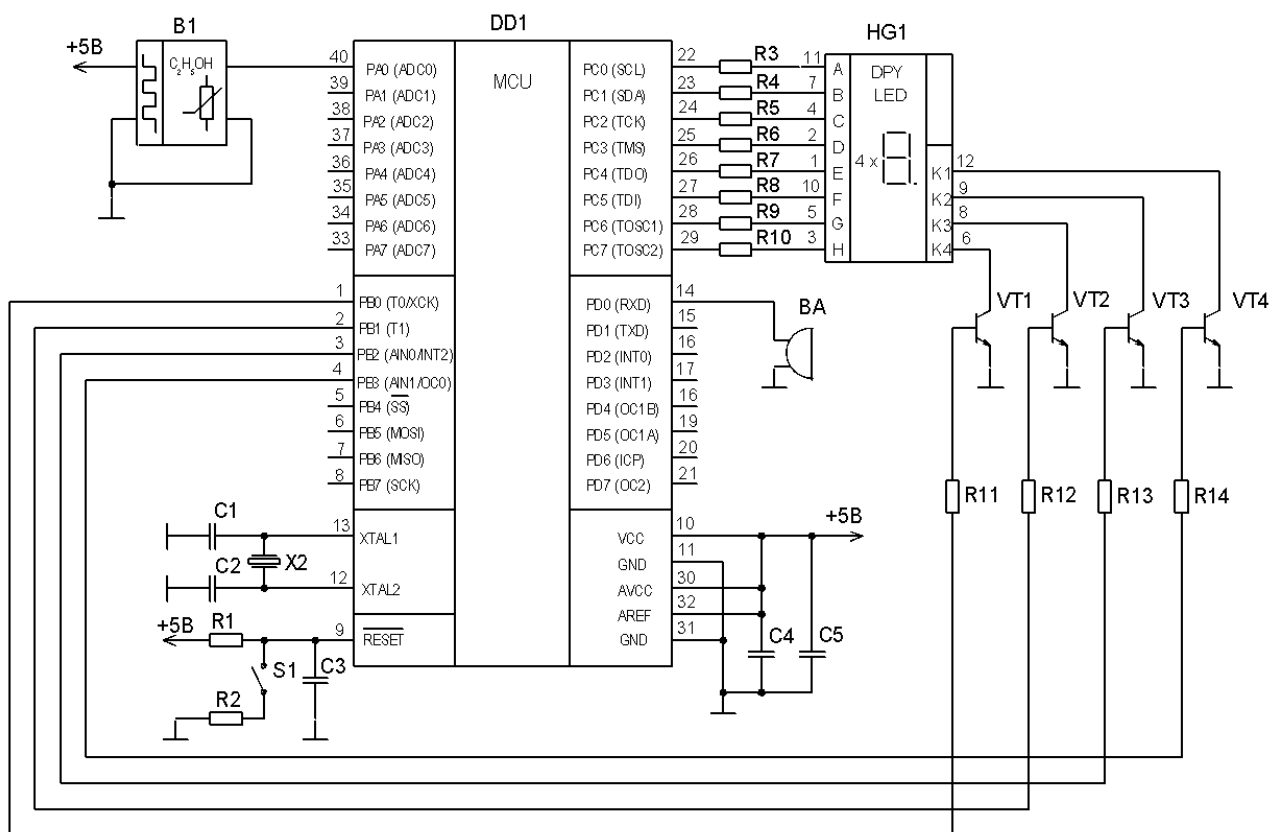


Рисунок 2.8 Принципиальная схема измерителя концентрации газа

3 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1 Измерение концентрации газовой примеси с помощью газоанализаторных датчиков серии MQ

Согласно справочным данным на используемый датчик MQ-3 [7], имеется экспериментальная зависимость отношения R_s/R_0 от концентрации газовой примеси в мг на литр воздуха. R_s — это сопротивление сенсора при исследовании концентрации примеси, R_0 — сопротивление сенсора при определённой концентрации паров спирта — 0,4 мг/л (рис. 3.1).

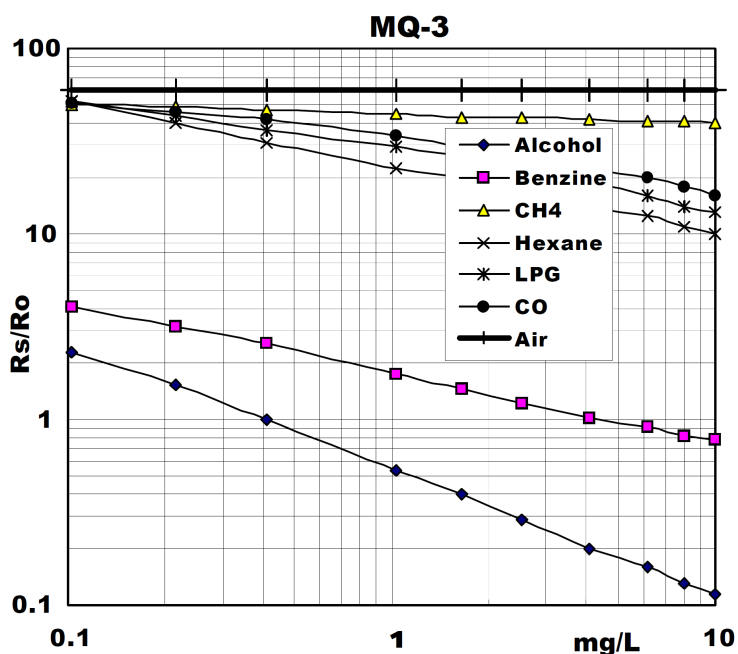


Рисунок 3.1 Зависимость отношения R_s/R_0 от концентрации различных газов датчика MQ-3

Таким образом проблема измерения концентрации сводится к решению трёх задач:

1. Нахождение значения R_0 или связанного с ним значения путем калибровки датчика после прогрева.
2. Определение отношения R_s/R_0 .
3. Нахождение по аппроксимирующей зависимости (рис. 3.1) концентрации газовой примеси (этанола).

Отношение сопротивления сенсора в чистом воздухе R_{Scl} к сопротивлению R_0 при отсутствии паров спирта заранее известно (см. рис. 3.1):

$$\frac{R_{Scl}}{R_0} = 60$$

Схема измерения и преобразования (рис. 3.2) дает возможность косвенным образом рассчитать при калибровке некоторые величины, которые затем можно использовать в процессе измерения. Отметим, что согласно справочному листку [7] сопротивление R_L заранее неизвестно (задаётся при подстройке на этапе изготовления датчика).

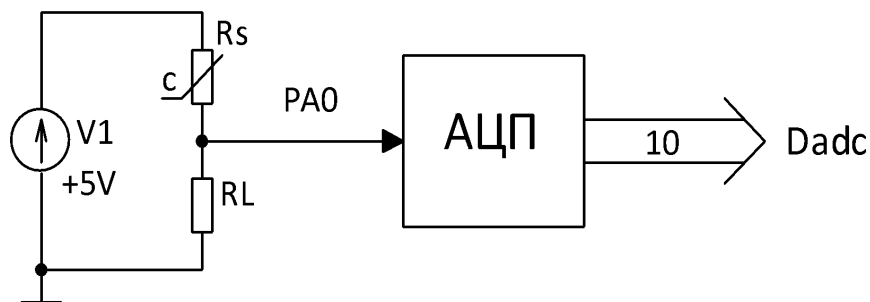


Рисунок 3.2 Схема измерения и преобразования данных

Получая преобразованное значение с АЦП в чистом воздухе (рис. 3.2) после прогрева датчика можно определить связь между сопротивлением сенсора в чистом воздухе R_{Scl} и сопротивлением R_L :

$$R_{Scl} = \frac{1024 - D_{ADCcl}}{D_{ADCcl}} \cdot R_L \quad (3.1)$$

Сопротивление сенсора при отсчетных условиях (концентрация этанола 0,4 мг/л):

$$R_0 = \frac{1024 - D_{ADC0}}{D_{ADC0}} \cdot R_L \quad (3.2)$$

В процессе измерения и обработки следует определить:

$$\frac{R_S}{R_0} = \frac{(1024 - D_{ADC}) \cdot D_{ADC0}}{D_{ADC} \cdot (1024 - D_{ADC0})} \cdot \frac{R_L}{R_L} = \frac{(1024 - D_{ADC}) \cdot D_{ADC0}}{D_{ADC} \cdot (1024 - D_{ADC0})} \quad (3.3)$$

Согласно (3.3) задача сводится к нахождению преобразованного кода напряжения датчика D_{ADC} при измерении концентрации. Отношение же $\frac{1024 - D_{ADC0}}{D_{ADC0}}$

можно определить заранее при калибровке датчика после прогрева (до захода в вечный цикл). Разделив (3.1) на (3.2) можно получить заранее известное отношение, равное 60:

$$\frac{R_{Scl}}{R_0} = \frac{(1024 - D_{ADCcl}) \cdot D_{ADC0}}{D_{ADCcl} \cdot (1024 - D_{ADC0})} = 60$$

Отсюда:

$$N_0 = \frac{1024 - D_{ADC0}}{D_{ADC0}} = \frac{1024 - D_{ADCcl}}{60 \cdot D_{ADCcl}}$$

Таким образом задача калибровки датчика состоит в нахождении кода АЦП напряжения с датчика в чистом воздухе D_{ADCcl} и затем в расчёте по нему значения N_0 по формуле:

$$N_0 = \frac{1024 - D_{ADCcl}}{60 \cdot D_{ADCcl}} \quad (3.4)$$

Затем в процессе циклических измерений в вечном цикле отношение R_S/R_0 определяется по формуле (получаемой подстановкой (3.4) в (3.3)):

$$\frac{R_S}{R_0} = \frac{(1024 - D_{ADC})}{D_{ADC} \cdot N_0} \quad (3.5)$$

Далее по найденному отношению R_S/R_0 следует определить концентрацию спирта. Отметим что логарифм отношения R_S/R_0 зависит линейно от логарифма концентрации (согласно рис. 3.1).

Эту зависимость можно (используя синюю линию рис. 3.1) ввести в таблицу Excel, построив её график (рис. 3.3). Видно, что зависимость сильно нелинейная. Далее можно построить другую таблицу и графическую зависимость, взяв натуральные логарифмы от концентрации и отношения

сопротивлений R_s/R_0 (рис. 3.4). Эта зависимость, естественно будет повторять зависимость рис. 3.1 для паров спирта. Но цель в другом. Используя возможность Excel находить аппроксимации экспериментальных зависимостей (построение линии тренда), можно найти для примерно линейной зависимости рис. 3.4 уравнение прямой линии $\ln(R_s/R_0) = A \cdot \ln(\text{conc}) + B$.

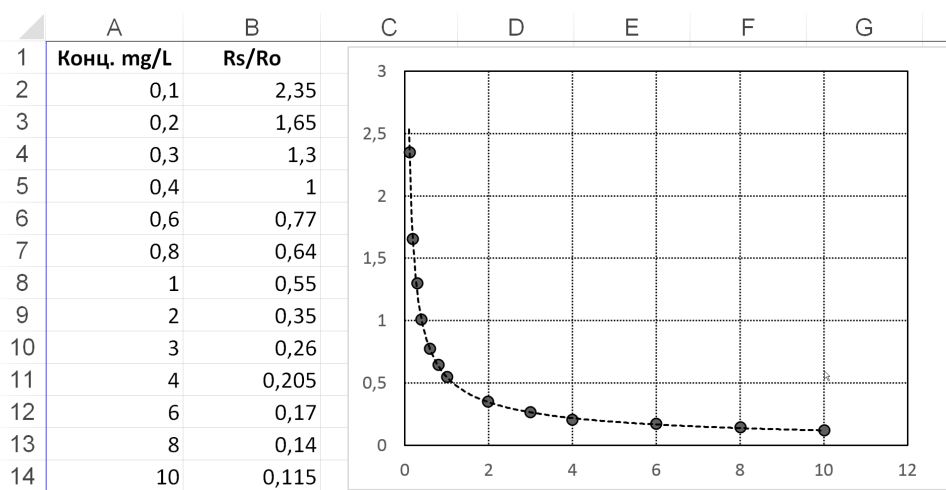


Рисунок 3.3 Зависимость отношения сопротивлений R_s/R_0 от концентрации

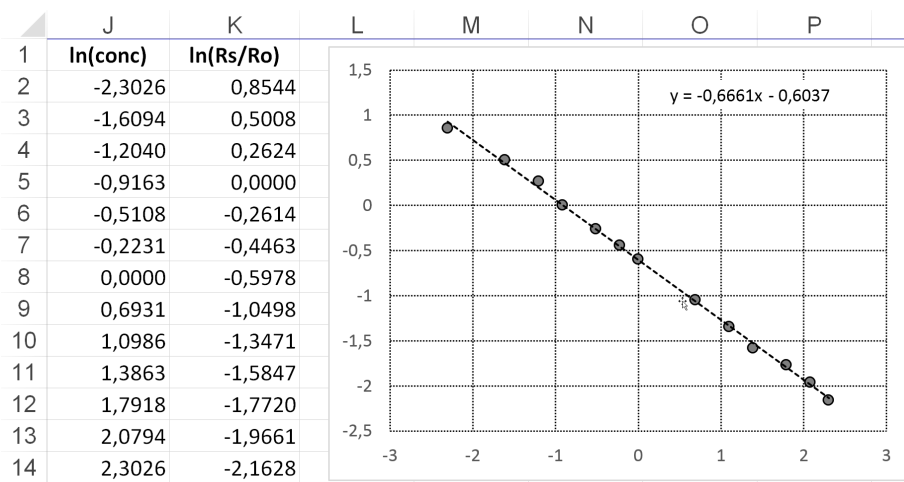


Рисунок 3.4 Зависимость логарифма отношения сопротивлений R_s/R_0 от логарифма концентрации примеси

Итак с помощью построения линии тренда в Excel получено значение коэффициентов A и B зависимости $\ln(R_s/R_0) = A \cdot \ln(\text{conc}) + B$. Отсюда формула для определения концентрации спирта в выдыхаемом воздухе в мг/литр:

$$conc = e^{\left(\frac{\ln\left(\frac{R_s}{R_0}\right) - B}{A} \right)} \quad [\text{мг/л}].$$

Для определения концентрации в промилле (г спирта на литр крови) вводится поправочный коэффициент 2,2:

$$conc = 2,2 \cdot e^{\left(\frac{\ln\left(\frac{R_s}{R_0}\right) - B}{A} \right)} \quad [‰].$$

Коэффициенты $A = -0,6661$, $B = -0,6037$ (см. на рис. 3.4 уравнение линии тренда).

3.2 Управление динамической индикацией

Импульсы тактового генератора микроконтроллера (частотой 12 МГц) делятся предделителем таймера на 256, таким образом получаются его тактовые импульсы частотой 46,875 кГц. Для получения частоты развертки изображения на семисегментном индикаторе $f_{\text{РАЗВ}} = 100$ Гц ($f_{\text{РАЗВ}} > f_{\text{КЧМ}}$) требуется задать время между прерываниями от таймера:

$$T_{3H} = \frac{1}{4} \cdot \frac{1}{f_{\text{РАЗВ}}} = \frac{1}{400} = 2,5 \text{ мс.}$$

Время счета таймера 0 в режиме CTC от нуля до вершины счета, записанной в регистре сравнения OCR0 (оно же и время между прерываниями по совпадению, если таковые разрешены):

$$T_{T0-CTC} = \frac{K_{PS}}{f_{OSC}} \cdot (OCR0 + 1),$$

где K_{PS} — коэффициент деления тактовой частоты предделителем таймера, f_{OSC} — тактовая частота работы микроконтроллера.

При известном коэффициенте предделителя ($K_{PS}=256$) и заданном времени между прерываниями $T_{T0-CTC}=T_{3H}=0,0025$ с, можно рассчитать требуемое значение в регистре сравнения OCR0:

$$OCR0 = \frac{T_{T0-CTC} \cdot f_{OSC}}{K_{PS}} - 1 = \frac{0,0025 \cdot 12 \cdot 10^6}{256} - 1 = 116$$

Таймер на соответствующий режим устанавливается в начальной части главного модуля до вхождения в вечный цикл (в инициализационной части программы), также в этом же месте записывается значение 116 в регистр OCR0. Далее значение в OCR0 остаётся неизменным.

В обработчике прерывания по совпадению организуется индикация текущего разряда числа на семисегментном индикаторе. Сначала выбирается индицируемое знакоместо путём включения соответствующего катодного ключа (запись логической 1 в соответствующий разряд младшей тетрады порта В). Затем из массива знакогенератора выбирается и выводится в порт С код графики цифры на данном знакоместе. Далее происходит проверка пройденного количества знакомест на >3 , при выполнении условия которой (цикл развертки по знакоместам закончен), номеру знакоместа присваивается нулевое значение (т.е. при следующем прерывании развёртка начнется со старшего разряда).

3.3 Блок-схема алгоритма

В соответствии со сказанным выше в подразделах 3.1 и 3.2 блок схема алгоритма будет иметь вид рис. 3.5.

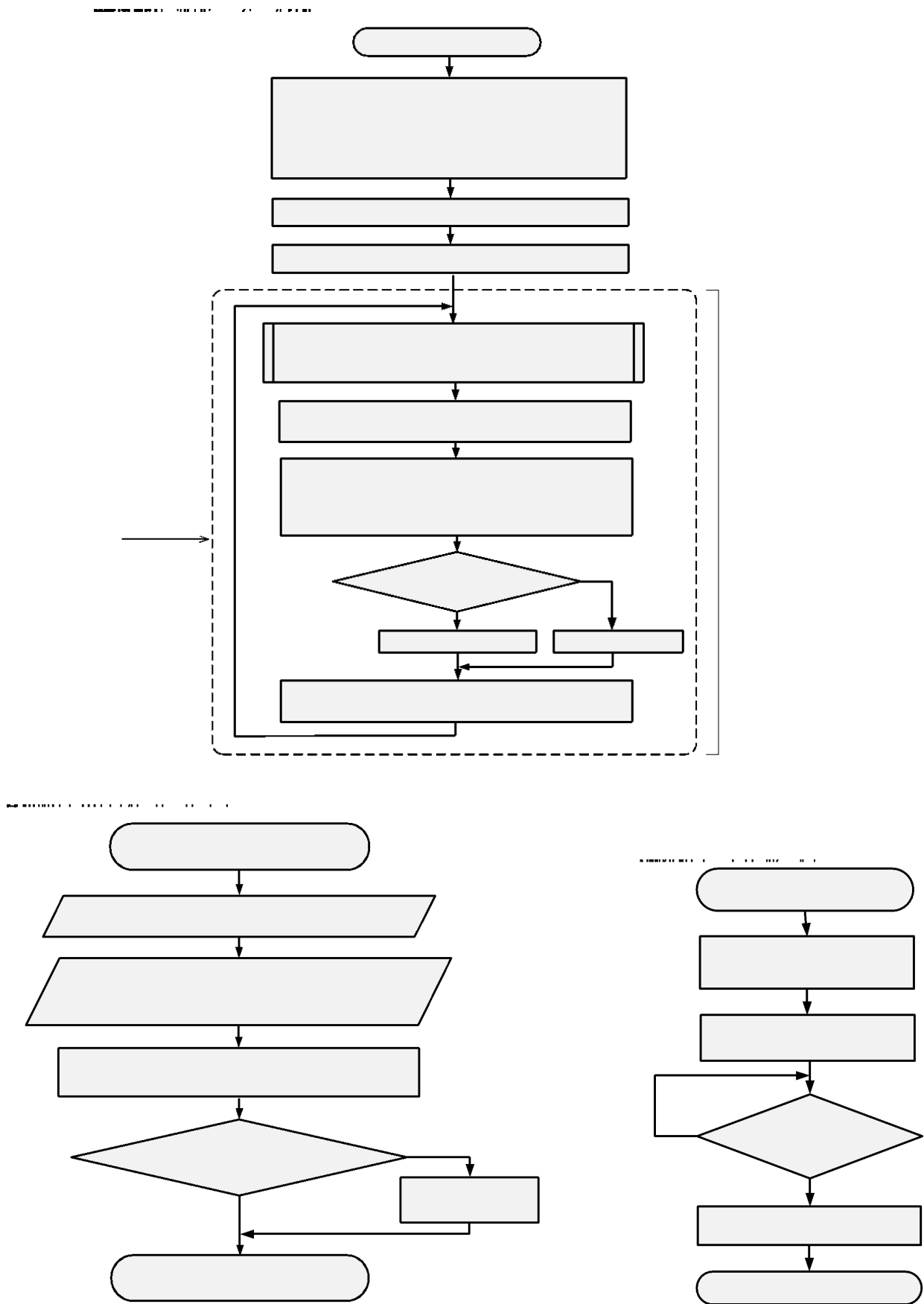


Рисунок 3.5 Блок-схема алгоритма измерения концентрации спирта

Основное время будет выполняться фоновая программа (бесконечный цикл, со считыванием преобразованных данных с АЦП МК, расчётом по ним концентрации, перевода значения концентрации в массив распакованных двоично-десятичных цифр. Этот бесконечный цикл будет периодически (через каждые 2,5 мс) прерываться обработчиком прерывания таймера по совпадению. Этот модуль организует индикацию текущего знакоместа на семисегментном индикаторе и организует циклическую смену индицируемых знакомест.

3.4 Программа на языке СИ

```

/*****
Chip type           : ATmega16
Program type        : Application
AVR Core Clock frequency: 12,000000 MHz
Memory model        : Small
External RAM size    : 0
Data Stack size     : 256
Программа определения концентрации паров алкоголя в
окружающем воздухе в мг/литр или для выдыхаемого воздуха - в промилле на литр
крови (по выбору). При превышении 0.25 промилле - зуммер
*****/
#include <mega16.h>
#include <delay.h>
#include <math.h>
//Declare your global variables here
//Библиотека графики 10 десятичных цифр (знакогенератор) для 7SEG с общим катодом
//////////0////////1////////2////////3////////4////////5////////6////////7////////8////////9////////H//
flash char DIG_DEC[11]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x76};
//число в распаке. BCD (одна цифра в одном байте), ст. байт по младшему адресу
//количество тысячных долей промилле, в исх. состоянии - индикация нагрева "HHHH"
unsigned char BCD_UNPACKED[4]={10, 10, 10, 10};
//индекс элементов массива распакованных десятичных цифр, номер индицируемого
//знака, начиная со старшего разряда - первого слева
unsigned char N_znaka=0;
// Voltage Reference: AVCC pin - Опорное напряжение для АЦП с вывода AVcc
#define ADC_VREF_TYPE ((0<<REFS1) | (1<<REFS0) | (0<<ADLAR))
//Подпрограмма чтения данных с АЦП, преобразующего напряжения заданного канала
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input | ADC_VREF_TYPE;
//Задержка требуется для стабилизации напряжения на входе АЦП (с выхода ADMUX)
delay_us(10);
//Запуск АЦП на преобразование
ADCSRA|=(1<<ADSC);
//Опрос готовности данных, ожидание пока преобразование не завершится
while ((ADCSRA & (1<<ADIF))==0);
ADCSRA|=(1<<ADIF); //Сброс флага готовности для возможности дальнейших преобр.
return ADCW; //возвращаемое функцией значение - содержимое регистров ADCH:ADCL
}

//Подпрограмма (обработчик прерывания по совпадению Timer0) смены знакомест 7SEG
interrupt [TIMO_COMP] void timer0_comp_isr(void)
{
/*Обнулять здесь TCO не надо - он сам сбросился при совпадении
и генерации прерывания (режим CTC)*/
PORTB=0x00; //Гашение текущего знакоместа, необходимо для модели
PORTC=0x00; //PROTEUS, для реального индикатора необязательно
PORTB=0b1000>>N_znaka; //Включаем катодный ключ очередного знакоместа
}

```

```

//Выводим в PORTC код графики знака 7-сегм. дисплея для очередного знакоместа
PORTC=DIG_DEC[BCD_UNPACKED[N_znaka++]]; //по окончании переводим указатель на след.
//десятичную цифру в массиве распакованных цифр
if (N_znaka==1) PORTC|=0x80; //если на 1-ом знакоместе, подсвечиваем точку
if (N_znaka>3) N_znaka=0; //Если обработан последний элемент массива,
//начинаем новый цикл
}

void main(void)
{
// Declare your local variables here
const float RATIO_CL=60.; //Отношение сопротивления сенсора Rs к Ro
// в чистом воздухе, Ro - сопротивление сенсора при определённой концентрации
// примеси в воздухе (для этанола при 0.4 мг/литр)*/
const float A=-0.6661, B=-0.6037; //множитель и смещение линейной аппроксимации
// логарифмической зависимости ln(Rs/Ro) от ln(концентрации): ln(Rs/Ro)=A*ln(конц)+B */
unsigned int i, adc;
float No; //No=Ro/RL - отношение сопр. сенсора при конц. 0.4 мг/л к сопр. нагр.*/

// Port B initialization Младшая тетрада - на вывод (базы катодных ключей)
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=Out Bit2=Out Bit1=Out Bit0=Out
DDRB=(0<<DDB7)|(0<<DDB6)|(0<<DDB5)|(0<<DDB4)|(1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
PORTB=(0<<PORTB7)|(0<<PORTB6)|(0<<PORTB5)|(0<<PORTB4)|(0<<PORTB3)|(0<<PORTB2)|
(0<<PORTB1)|(0<<PORTB0);

// Port C initialization. Весь порт C - на вывод данных (8 сегментов)
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
DDRC=(1<<DDC7)|(1<<DDC6)|(1<<DDC5)|(1<<DDC4)|(1<<DDC3)|(1<<DDC2)|(1<<DDC1)|(1<<DDC0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
PORTC=(0<<PORTC7)|(0<<PORTC6)|(0<<PORTC5)|(0<<PORTC4)|(0<<PORTC3)|(0<<PORTC2)|
(0<<PORTC1)|(0<<PORTC0);

// Port D initialization. Младший бит PORTD.0 - на вывод (зуммер)
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=Out
DDRD=(0<<DDD7)|(0<<DDD6)|(0<<DDD5)|(0<<DDD4)|(0<<DDD3)|(0<<DDD2)|(0<<DDD1)|(1<<DDD0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTD=(0<<PORTD7)|(0<<PORTD6)|(0<<PORTD5)|(0<<PORTD4)|(0<<PORTD3)|(0<<PORTD2)|
(0<<PORTD1)|(0<<PORTD0);

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 46,875 kHz
// Mode: CTC top=OCR0
// OCO output: Disconnected
// Timer Period: 2,496 ms
//запуск таймера с частотой Fclk/256=46,875kHz в режиме CTC
TCCR0=(0<<WGM00)|(0<<COM01)|(0<<COM00)|(1<<WGM01)|(1<<CS02)|(0<<CS01)|(0<<CS00);

```

```

TCNT0=0x00;
OCR0=116;  /*загрузка регистра сравнения числом, обеспечивающем счет
от нуля до совпадения в течение 2.5 мс (частота развертки 100Гц) */
//разрешение прерывания по совпадению TC0
TIMSK=(0<<OCIE2)|(0<<TOIE2)|(0<<TICIE1)|(0<<OCIE1A)|(0<<OCIE1B)|(0<<TOIE1)|(1<<OCIE0)|
(0<<TOIE0);

// ADC initialization
// ADC Clock frequency: 187,500 kHz - это частота преобразования
// ADC Voltage Reference: AVCC pin - опорное напряжение берётся с аналогового питания
// ADC Auto Trigger Source: ADC Stopped
ADMUX=ADC_VREF_TYPE;
ADCSRA=(1<<ADEN)|(0<<ADSC)|(0<<ADATE)|(0<<ADIF)|(0<<ADIE)|(1<<ADPS2)|(1<<ADPS1)|
(0<<ADPS0);
SFIOR=(0<<ADTS2)|(0<<ADTS1)|(0<<ADTS0);

asm("sei")          //Глобальное разрешение прерываний

/*Прогрев датчика в течение 5 минут, индикация НННН - идёт подготовка
к измерению, прогрев датчика*/
for (i=0; i<30; i++) delay_ms(10000);          //для 5-минутного прогрева i<30

/*Калибровка датчика - нахождение отношения сопротивлений Ro/RL=No (Ro - сопротивление
сенсора при конц. 0.4 мг/л) при текущих условиях: температуре, влажности, давлении,
напряжении на нагревателе датчика в чистом воздухе (при отсутствии паров спирта)*/

adc=read_adc(0);    //чтение данных с АЦП в переменную adc

/*Вычисление отношения No=Ro/RL при отсутствии паров спирта в воздухе*/
No=((1024.- (float)adc)/(float)adc)/RATIO_CL;
/*Вывод показания АЦП в чистом воздухе при прогревом сенсоре*/
for (i=0; i<=3; i++) BCD_UNPACKED[i]=0; //обнуление массива, т.к. неизв. число знач. цифр
/*нахождение распак. десятичных цифр числа adc и заполнение массива BCD_UNPACKED[4]*/
for (i=3; adc>0; i--, adc=adc/10) BCD_UNPACKED[i]=adc%10;
/*прерывистый 10 сек зуммер при показе кода АЦП в чистом воздухе, после его окончания
можно осуществлять измерение*/
for (i=0; i<10; i++)
    {PORTD=1, delay_ms(500); PORTD=0, delay_ms(500)};

/*циклическое нахождение концентрации алкоголя в выдыхаемом воздухе в промилле на литр*/
while (1)
{
    // Place your code here
    float ratio, per_mille;
    unsigned int per_micro;

```

```

adc=read_adc(0);
/*вычисление отношения сопр. Rs/Ro для использования в аппроксимирующей функции*/
ratio=((1024.- (float)adc)/(float)adc)/No;
/* вычисление конц. этанола в мг/литр воздуха */
per_mille=exp((log(ratio)-B)/A);          //вычисление по аппрокс. формуле
/*если нужен вывод в мг на литр воздуха, то следующую строку закомментировать*/
per_mille=2.2*per_mille; /*вычисление в промиллях (г спирта на литр крови)*/
per_micro=(unsigned int)(per_mille*1000); //получение целого числа тысячных долей
if (per_micro>=250) {PORTD=1;}             //если более 0,25 промилле включаем зуммер
    else {PORTD=0;};                       //если нет молчим
/*заполнение массива цифр нулями, т.к. заранее неизвестно кол-во значащих цифр*/
for (i=0; i<=3; i++) BCD_UNPACKED[i]=0;
//нахождение распак. десят. цифр per_micro и заполнение массива BCD_UNPACKED[4]
for (i=3; per_micro>0; i--, per_micro=per_micro/10) BCD_UNPACKED[i]=per_micro%10;
delay_ms(300);
}
}

```

4 ПРОВЕРКА РАБОТОСПОСОБНОСТИ УСТРОЙСТВА

4.1 Моделирование в среде PROTEUS

Для моделирования в среде PROTEUS, сначала в среде IDE CodeVisionAVR создается проект с использованием встроенного генератора предварительного кода Code Wizard AVR (мастер настройки программного кода). При работе в мастере предварительного кода на первом этапе устанавливаются такие свойства устройства как тип МК, его тактовая частота (в данном проекте 12 МГц). На втором этапе производятся настройки периферии МК (таймера TCNT0 и АЦП) в режим, соответствующий инициализационной части программы, разрешаются необходимые прерывания (по совпадению TCNT0).

После этого может быть сгенерирован предварительный код, который затем корректируется в соответствии с используемой блок-схемой алгоритма: заполняются кодом на СИ обработчики прерываний, пишутся фрагменты, обеспечивающие обработку результатов и вывод их на индикацию. В результате получается программа (см. подраздел 3.4), которая затем компилируется в среде данной IDE.

Продуктами компиляции (после исправления всех синтаксических ошибок) являются:

- объектный файл (с расширением *cof*) — он затем используется для эмуляции и отладки проекта в среде PROTEUS;
- бинарный файл (с расширением *hex*) — эти данные заносятся во flashпамять МК на этапе проверки работы проекта на отладочной плате через программатор USBASP.

В графическом редакторе системы PROTEUS рисуется принципиальная схема проекта (рис. 4.1), производятся установки для микроконтроллера (рис. 4.1) в соответствии с выбранным режимом тактирования, подсоединяется

объектный файл (рис. 4.1). Далее производится отладка программы, в результате которой выявляются логические ошибки.

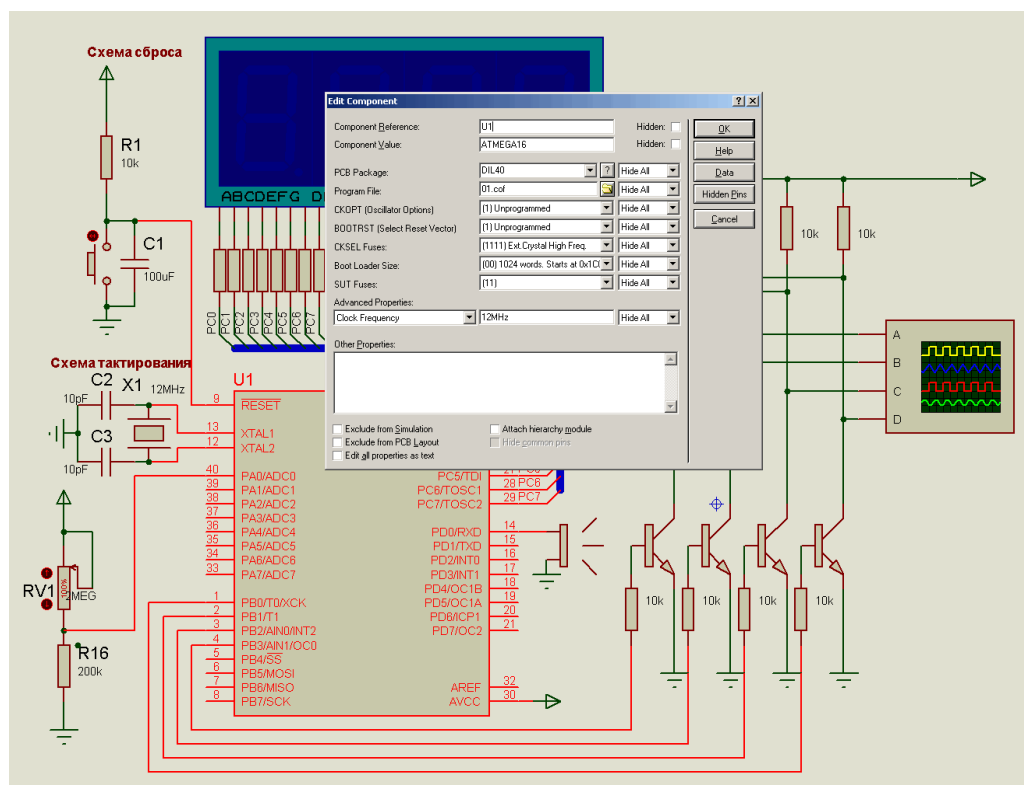


Рисунок 4.1 Схема для моделирования и установки свойств МК

После исправления логических ошибок в IDE Code Vision AVR, производится итоговая компиляция программы и далее можно проводить моделирование работы измерителя при разных величинах сопротивления верхнего плеча делителя, моделирующего сопротивление сенсора датчика (рис. 4.3–4.5).

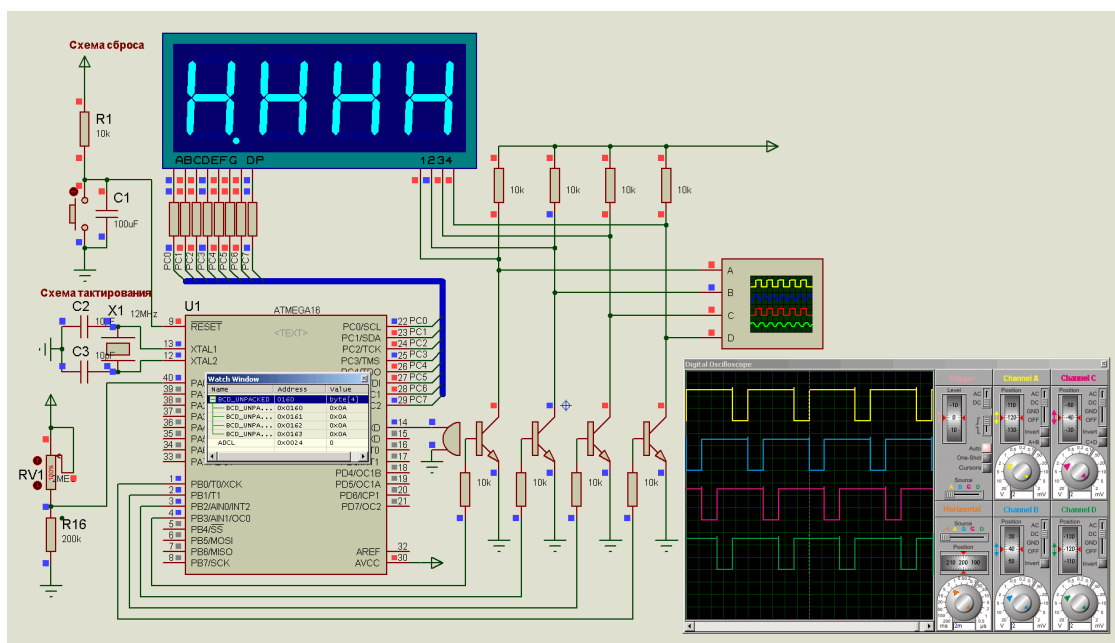


Рисунок 4.2 Прогрев датчика

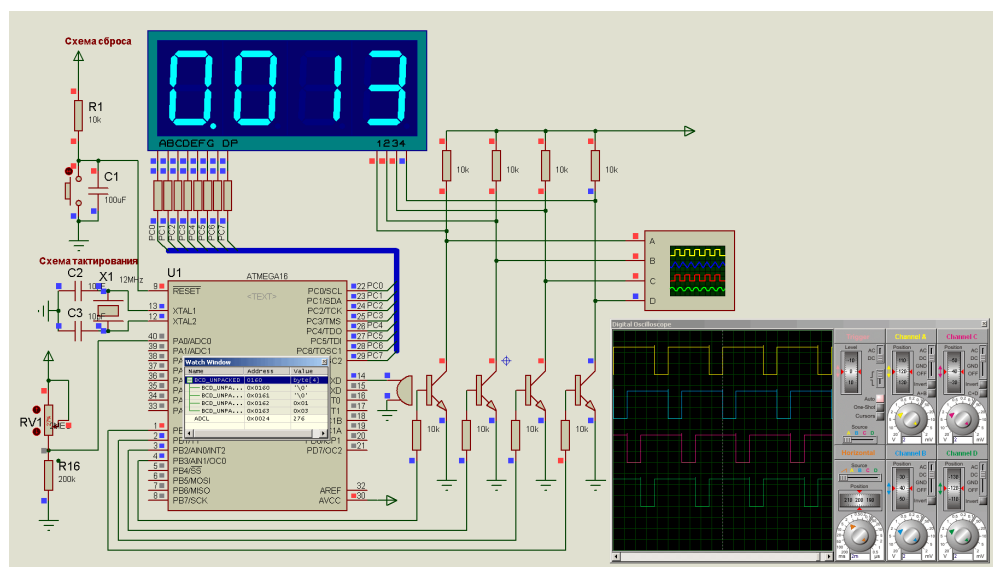


Рисунок 4.3 Малая концентрация спирта

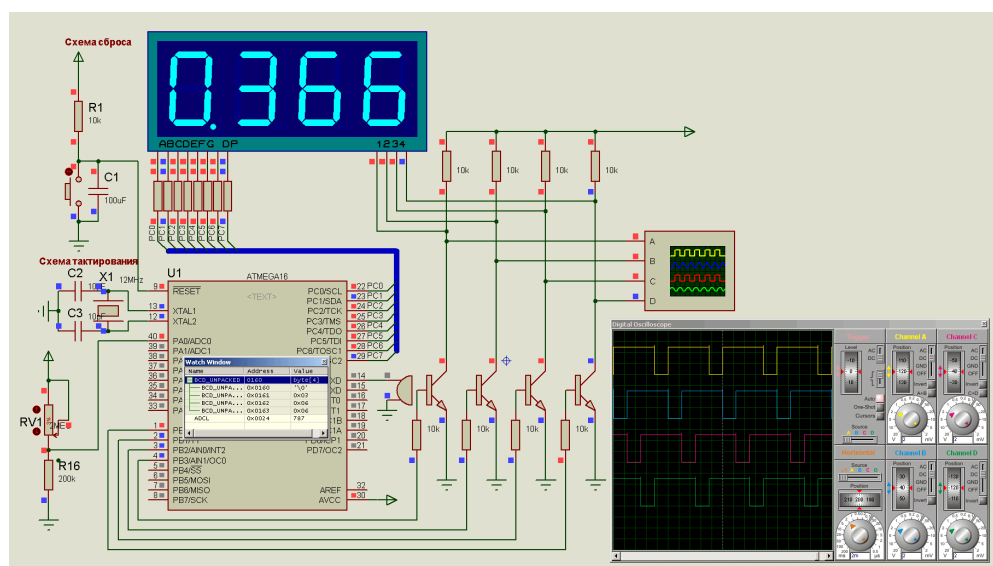


Рисунок 4.4 Существенная концентрация спирта

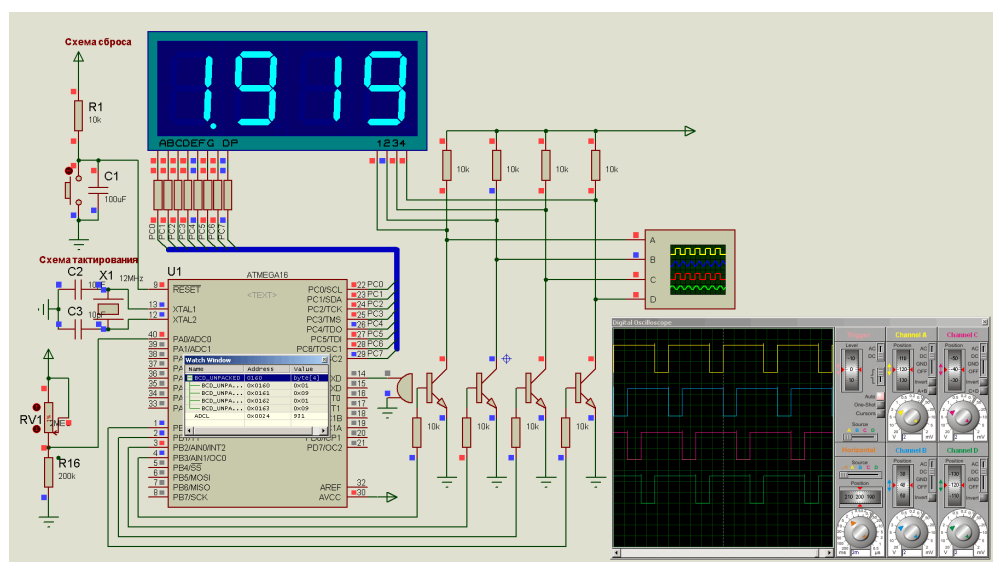


Рисунок 4.5 Максимальная концентрация спирта

Таким образом, моделирование в среде PROTEUS (с параметрами резистивного делителя, близкими к реальным значениям внутри датчика) показало, что устройство работает в соответствии с техническим заданием.

4.2 Проверка работы измерителя концентрации спирта с помощью отладочной платы

Для совместного программно-аппаратного тестирования собирается электронная система, состоящая из простой отладочной платы на основе МК

ATmega16, датчика MQ-3, модуля индикации, зуммера и подсоединяемого через разъем источника питания +5V (рис. 4.6).

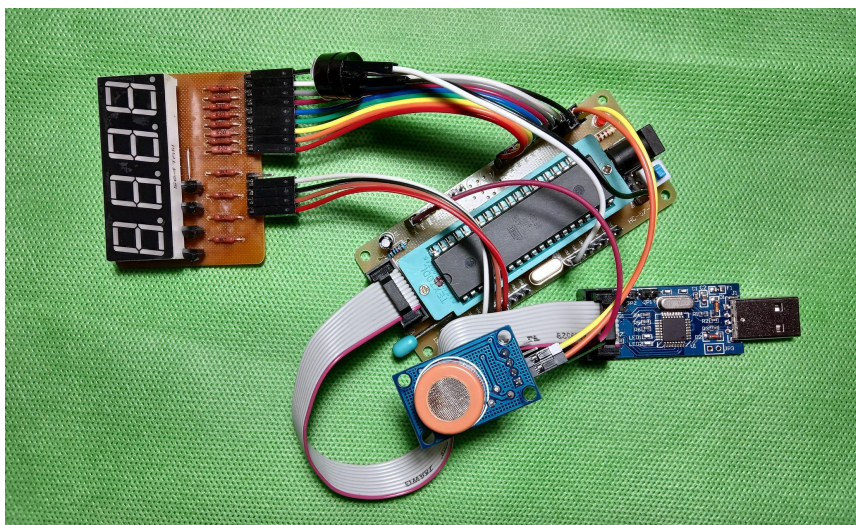


Рисунок 4.6 Система для совместного программно-аппаратного тестирования устройства

Flash-память МК Atmega16 отладочной платы предварительно программируется двоичным кодом оттранслированной программы (файлом с расширением *.hex, являющегося одним из продуктов компиляции программы IDE Code VisionAVR) с помощью программы AVRDUDE_PROG 3.3 и программатора USBASP.

Программатор USBASP с подсоединенной отладочной платой подключается к ПК через разъем USB. Через него же она получает необходимое питающее напряжение 5 В и через этот интерфейс производится прошивка Flashпамяти микроконтроллера бинарным кодом программы.

Ниже на рис. 4.7–4.9 показаны фотографии экспериментов в течение прогрева датчика, определения концентрации в чистом воздухе и определение концентрации при наличии паров спирта.

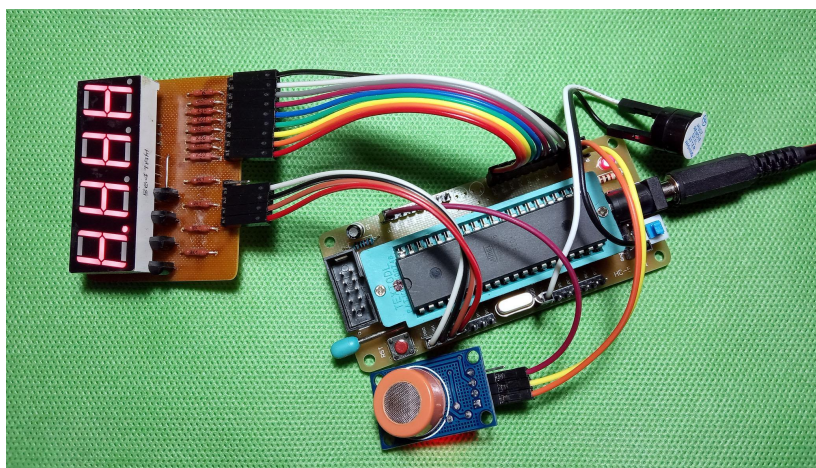


Рисунок 4.7 Отладочная плата сразу после запуска программы (прогрев)

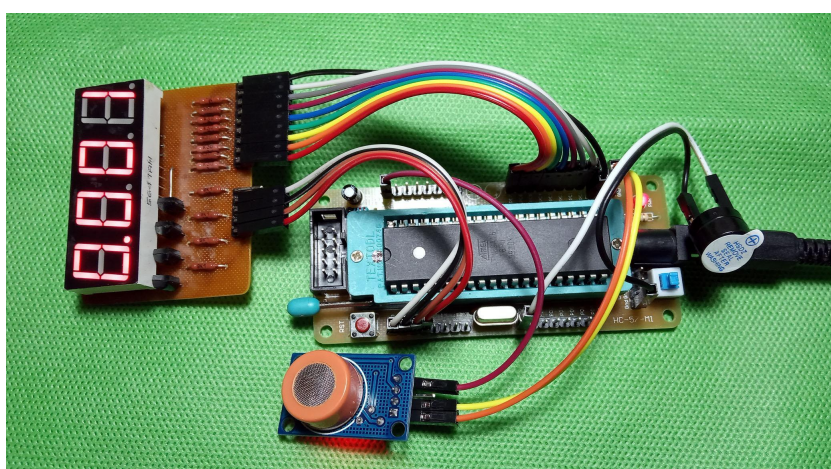


Рисунок 4.8 Измерение концентрации спирта в чистом воздухе

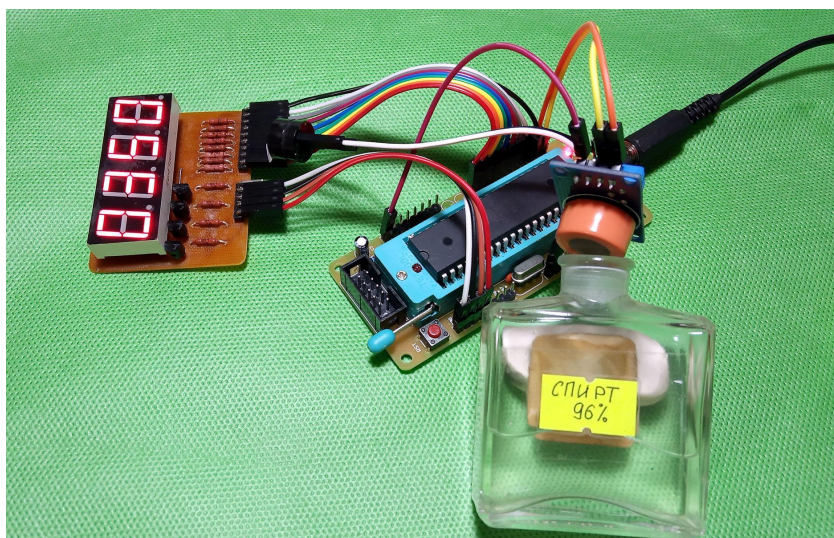


Рисунок 4.9 Измерение концентрации этанола при наличии спирта

Эксперимент показал, что измерения производятся правильно.

ЗАКЛЮЧЕНИЕ

При выполнении курсового проекта была произведена разработка измерителя концентрации спирта либо в мг/л в выдыхаемом воздухе либо содержание в крови в промилле. Измеритель концентрации выполнен на основе микроконтроллера ATmega16.

Приведено подробное описание реализации устройства на уровне структурной и функциональной схем. Разработана блок-схема алгоритма работы измерителя концентрации, в соответствии с ней написана управляющая программа для микроконтроллера на языке СИ. Получен код прошивки для МК, позволивший провести тестирование устройства с помощью отладочной платы.

Разработан необходимый комплект конструкторской документации для изготовления макетного образца изделия: принципиальная схема и перечень элементов.

Осуществлена проверка работоспособности устройства как с помощью имитационного моделирования в среде пакета PROTEUS, так и путем совместного программно-аппаратного тестирования с использованием отладочной платы.

Моделирование и натурный эксперимент показали, что устройство работоспособно и его параметры соответствуют техническому заданию.

СПИСОК ИСТОЧНИКОВ

1. Амелина М.А., Троицкий Ю.В. Программирование микроконтроллеров семейства AVR в устройствах промышленной электроники на языке Си. Лабораторный практикум в 2-х частях по курсу «Основы микропроцессорной техники». Часть 1. Смоленск: РИО филиала МЭИ в г. Смоленске, 2015. 56 с.
2. Абраменкова И.В., Семченков Н.С., Троицкий Ю.В. Применение языка Си для программирования микроконтроллеров семейства AVR в устройствах промышленной электроники. Учебное пособие по курсу «Основы микропроцессорной техники». Смоленск: Филиал ГОУВПО «МЭИ (ТУ)» в г. Смоленске, 2007. 84 с.
3. Лебедев М.Б. CodeVisionAVR: пособие для начинающих. М.: Додэка-XXI, 2008. 592 с.
4. EasyAVR5A user manual. MikroElektronika. Software and hardware solution for embedded world. V101. URL: http://www.mikroe.com/downloads/get/28/easyavr5a_manual_v101.pdf (дата обращения: 10.01.2016).
5. Амелина М.А., Троицкий Ю.В. Программирование микроконтроллеров семейства AVR в устройствах промышленной электроники на языке Си. Лабораторный практикум в 2-х частях по курсу «Основы микропроцессорной техники». Часть 2. Смоленск: РИО филиала МЭИ в г. Смоленске, 2015. 108 с.
6. Евстифеев А.В. Микроконтроллеры AVR семейств Tiny и Mega фирмы ATMEL, 5-е изд., стер. М.: Издательский дом «Додэка-XXI», 2008. 560 с.
7. Technical Data MQ-3 Gas Sensor (Электронный ресурс). URL: <https://www.sparkfun.com/datasheets/Sensors/MQ-3.pdf> (Дата обращения 23.12.2018)

Схема электрическая принципиальная

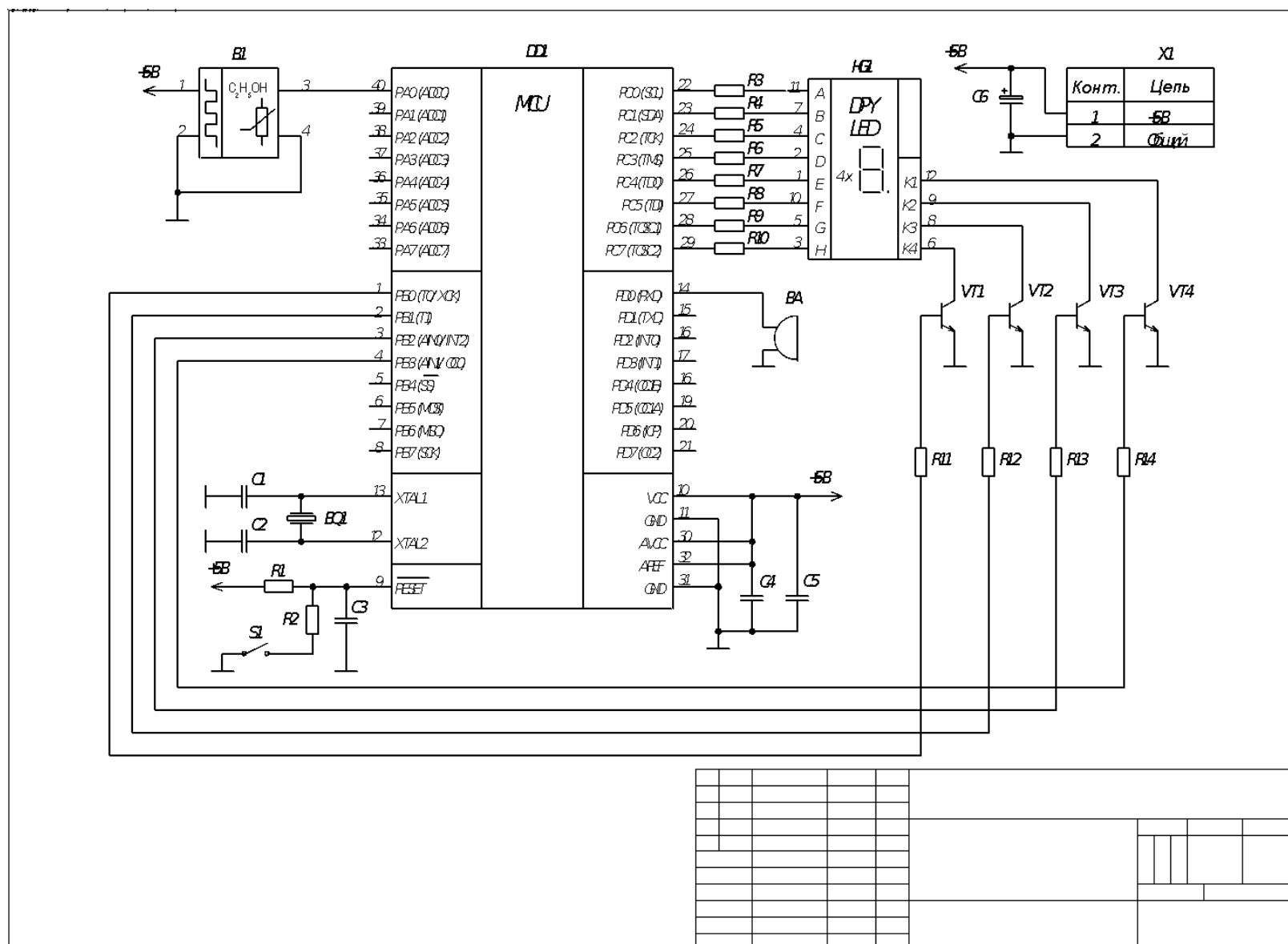


Рисунок А.1 Уменьшенная копия чертежа принципиальной электрической схемы

ПРИЛОЖЕНИЕ 2

Перечень элементов

