



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Отчет по курсовой работе

«Разработка цифрового вольтметра»

по дисциплине
Микропроцессорные системы

РУКОВОДИТЕЛЬ:

(подпись)

Киселёв Ю.Н.
(фамилия, и.,о.)

СТУДЕНТ:

(подпись)

Сухоруков В. А
(фамилия, и.,о.)

19-В-2
(шифр группы)

Работа защищена «__» _____
С оценкой _____

Нижний Новгород 2022

Оглавление

| | | |
|--------|---|----|
| 1. | Введение | 3 |
| 1.1. | Цели и задачи работы. | 3 |
| 1.2. | Состояние вопроса. | 3 |
| 1.2.1. | Структурная схема вольтметра | 3 |
| 1.2.2. | Делитель напряжения..... | 4 |
| 1.2.3. | АЦП..... | 5 |
| 1.2.4. | Устройство управления..... | 6 |
| 1.2.5. | Вольтметр с автоматическим выбором пределов измерения..... | 6 |
| 2. | Разработка технического задания | 7 |
| 3. | Анализ технического задания и разработка концептуальной модели..... | 7 |
| 3.1. | Выбор метода решения задачи | 7 |
| 3.2. | Построение концептуальной модели системы..... | 10 |
| 4. | Разработка аппаратной части устройства..... | 11 |
| 4.1. | Выбор программного обеспечения для разработки..... | 11 |
| 4.2. | Выбор элементов и проектирование схемы | 11 |
| 4.3. | Описание микроконтроллера | 12 |
| 4.4. | Принципиальная схема устройства | 14 |
| 5. | Разработка программной части устройства | 15 |
| 5.1. | Выбор среды разработки | 15 |
| 5.2. | Алгоритм работы программы | 16 |
| 6. | Демонстрация работы устройства..... | 19 |
| 6.1. | Измерение напряжения в диапазоне от 0В до 250В | 19 |
| 6.2. | Измерение напряжения в диапазоне от 0В до 100В | 19 |
| 6.3. | Измерение напряжения в диапазоне от 0В до 10В | 20 |
| 6.4. | Измерение напряжения в диапазоне от 0В до 5В | 20 |
| 7. | Заключение. | 20 |
| 8. | Приложение 1. Исходный текст программы (листинг)..... | 21 |
| 9. | Приложение 2. Информационные источники. | 24 |

1. Введение

1.1. Цели и задачи работы.

- ❖ Изучить структурную схему, и принцип работы цифрового вольтметра.
- ❖ Выбрать компоненты для разработки и проектирования вольтметра.
- ❖ Спроектировать цифровой вольтметр в пакете программ Proteus.

1.2. Состояние вопроса.

1.2.1. Структурная схема вольтметра

Вольтметр - это прибор, используемый для измерения разности электрических потенциалов между двумя точками электрической цепи. Он подключен параллельно, и имеет высокое сопротивление, так что он берет незначительный ток из цепи.

Цифровой вольтметр (ЦВ) – это цифровой прибор, автоматически вырабатывающий дискретные сигналы измерительной информации, показания которых представляются в цифровой форме. В ЦВ в соответствии со значением измеряемого напряжения образуется код, а затем в соответствии с кодом измеряемая величина представляется на отсчетном устройстве в цифровой форме.

Упрощенная структурная схема ЦВ, состоящая из входного устройства (ВУ), аналогово-цифрового преобразователя (АЦП), цифрового отсчетного устройства (ЦОУ), управляющего устройства (УУ), приведена на рисунке 1.

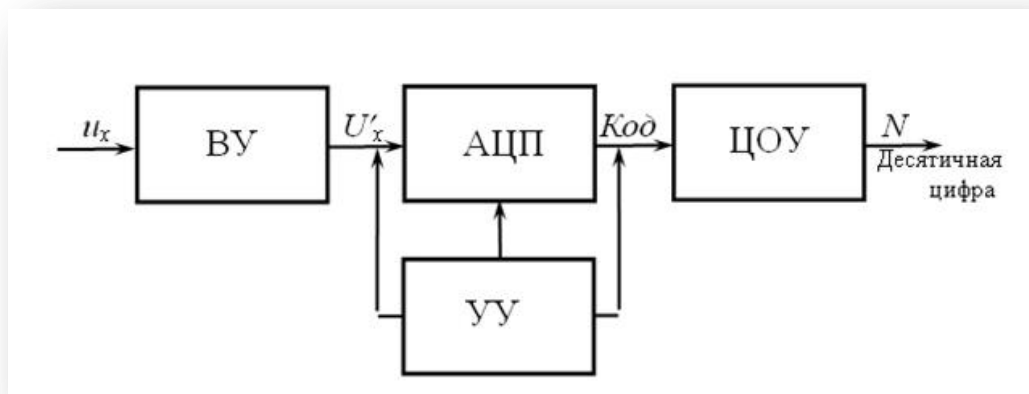


Рис 1. Структурная схема цифрового вольтметра

ВУ содержит делитель напряжения. АЦП преобразует аналоговый сигнал в цифровой, представляемый цифровым кодом. ЦОУ измерительного

прибора регистрирует измеряемую величину. УУ объединяет и управляет всеми узлами вольтметра.

1.2.2. Делитель напряжения

Схема делителя напряжения включает в себя входной источник напряжения и два резистора. На рисунке 2 представлено несколько схематических вариантов изображения делителя, но все они несут один и тот же функционал.

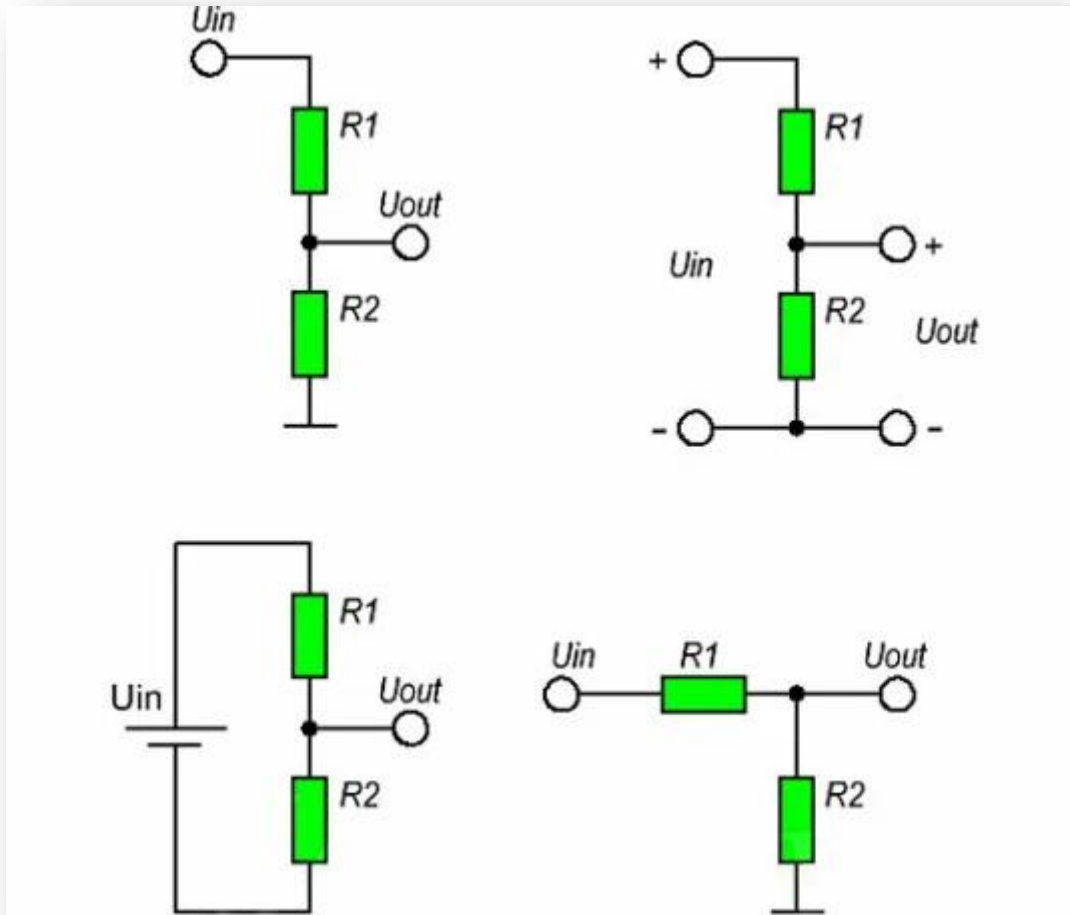


Рис 2. Структурная схема делителя напряжения

Обозначим резистор, который находится ближе к плюсу входного напряжения (U_{in}) как $R1$, а резистор находящийся ближе к минусу как $R2$. Падение напряжения (U_{out}) на резисторе $R2$ — это пониженное напряжение, полученное в результате применения резисторного делителя напряжения.

Расчет делителя напряжения предполагает, что нам известно, по крайней мере, три величины из приведенной выше схемы: входное напряжение и сопротивление обоих резисторов. Зная эти величины, мы можем рассчитать выходное напряжение.

Для того чтобы узнать какое напряжение будет на выходе делителя, выведем формулу исходя из закона Ома. Предположим, что мы знаем значе-

ния U_{in} , $R1$ и $R2$. Теперь на основании этих данных выведем формулу для U_{out} . Обозначим токи $I1$ и $I2$ - как токи, которые протекают через резисторы $R1$ и $R2$ соответственно.

$$U_{out} = I_2 \times R2$$

Рис 3.

Закон Ома для участка цепи в его наиболее простом виде: $U_{in} = I \times R$. R состоит из $R1+R2$. Формула может быть записана в следующем виде:

$$I = \frac{U_{in}}{R1 + R2}$$

Рис 4.

А так как $I1$ равно $I2$, то:

$$U_{out} = R2 \times \frac{U_{in}}{R1 + R2}$$

Рис 5. Формула выходного напряжения делителя напряжения

Это уравнение показывает, что выходное напряжение прямо пропорционально входному напряжению и отношению сопротивлений $R1$ и $R2$.

1.2.3. АЦП

АЦП микроконтроллера умеет измерять только напряжение. Чтобы произвести измерение других физических величин, их нужно вначале преобразовать в напряжение. Сигнал всегда измеряется относительно точки называемой опорное напряжение, эта же точка является максимумом, который можно измерить. В качестве источника опорного напряжения (ИОН), рекомендуется выбирать высокостабильный источник напряжения, иначе все измерения будут плясать вместе с опорным.

Одной из важнейших характеристик является разрешающая способность, которая влияет на точность измерения. Весь диапазон измерения разбивается на части. Минимум ноль, максимум напряжение ИОН. Для 8 битного АЦП это $2^8=256$ значений, для 10 битного $2^{10}=1024$ значения. Таким образом, чем выше разрядность, тем точнее можно измерять сигнал.

В качестве источника опорного напряжения можно использовать внутренний источник и внешний. Напряжение внутреннего источника (2,3-2,7В) не рекомендуется использовать, по причине низкой стабильности.

Внешний источник подключается к ножке AVCC или Aref, в зависимости от настроек программы.

1.2.4. Устройство управления

Устройство управления предназначено для управления работой аналогового-цифрового преобразователя, а также для формирования сигналов, поступающих на схему индикации.

1.2.5. Вольтметр с автоматическим выбором пределов измерения

Вольтметры, способные измерять напряжение в нескольких диапазонах, имеют в своей конструкции несколько делителей напряжения для более точного измерения в нужном диапазоне.

Каждый делитель «опускает» напряжение до того, которое способен преобразовать АЦП микро-эвм. В зависимости от входного напряжения на измеряемом приборе сигнал на АЦП подается с одного из нескольких делителей с разными коэффициентами.

2. Разработка технического задания

Разработать цифровой вольтметр с автоматическим выбором шкалы измерения на базе микро-ЭВМ.

Предел измерения напряжения постоянного тока от 0 до +250 в. Относительная погрешность измерения напряжения не хуже 0.1% от шкалы.

Диапазоны измерений:

1. 0в до +250в
2. 0в до +100в
3. 0в до +10в
4. 0в до +5в

Время измерения не более 0.2сек.

Использовать Протеус при разработке и моделировании.

3. Анализ технического задания и разработка концептуальной модели

3.1. Выбор метода решения задачи

В качестве микро-ЭВМ выберем модуль Arduino Uno (Рис 6). Данный модуль имеет 6 входов, способных работать в режиме АЦП, что позволит удобно считывать данные и делителей напряжения.

АЦП платы Arduino имеют разрядность 10 бит, поэтому значения на их выходах будут лежать в диапазоне от 0 до 1023, что позволит повысить точность измерения по сравнению с 8 битными АЦП, диапазон которых от 0 до 254.

Для вывода результата измерения будем использовать ЖК-дисплей размером 16x2 (16 символов в двух строках). В пакете Proteus данный дисплей представлен модель LM016L (Рис 7).

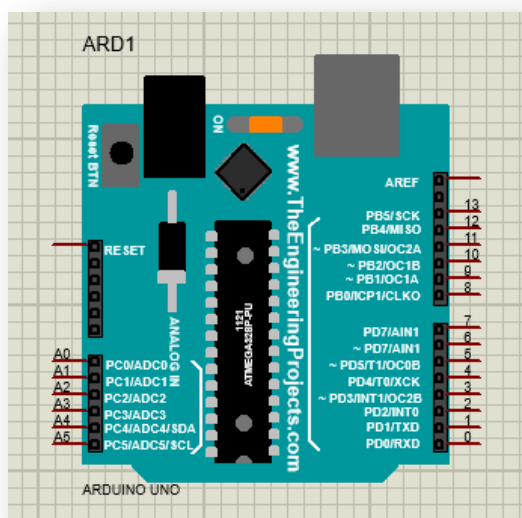


Рис 6. Модель платы Arduino Uno в пакете Proteus

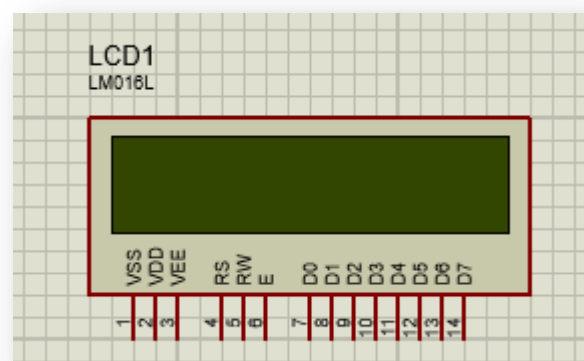


Рис 7. Модель ЖК дисплея LM016L в пакете Proteus

АЦП платы Arduino Uno может преобразовывать напряжение от 0В до 5В. Для измерения напряжения в разных диапазонах будем использовать делители напряжения с разными коэффициентами:

1) Диапазон от 0В до 5В. Напряжение изменять не нужно, делитель не требуется. Установим один резистор R1 сопротивлением 220 Ом.

2) Диапазон от 0В до 10В. Необходимо уменьшить напряжение в 2 раза. Рассчитаем номиналы сопротивления для резистивного делителя напряжения.

$$\frac{10}{R2 + R3} * R3 = 5$$

$$2 * R3 = R3 + R2$$

$$R3 = R2$$

Сопротивления резисторов должны быть равными. Выберем резисторы с сопротивлением 10кОм.

3) Диапазон от 0В до 100В. Необходимо уменьшить напряжение в 20 раз.

$$\frac{100}{R4 + R5} * R5 = 5$$

$$20 * R5 = R4 + R5$$

Возьмём резистор R4 сопротивлением 25кОм, тогда сопротивление резистора R5 должно быть равно 1315 Ом.

4) Диапазон от 0В до 250В. Необходимо уменьшить напряжение в 50 раз.

$$\frac{250}{R6 + R7} * R7 = 5$$

$$50 * R7 = R7 + R6$$

Возьмём резистор R6 сопротивлением 50кОм, тогда сопротивление резистора R7 должно быть равно 1020 Ом.

Для переключения направления тока от делителей напряжения на АЦП будем использовать Мультиплексор. Переключение источника данных будет изменяться подачей кодового слова на управляющие входы мультиплексора от платы Arduino Uno.

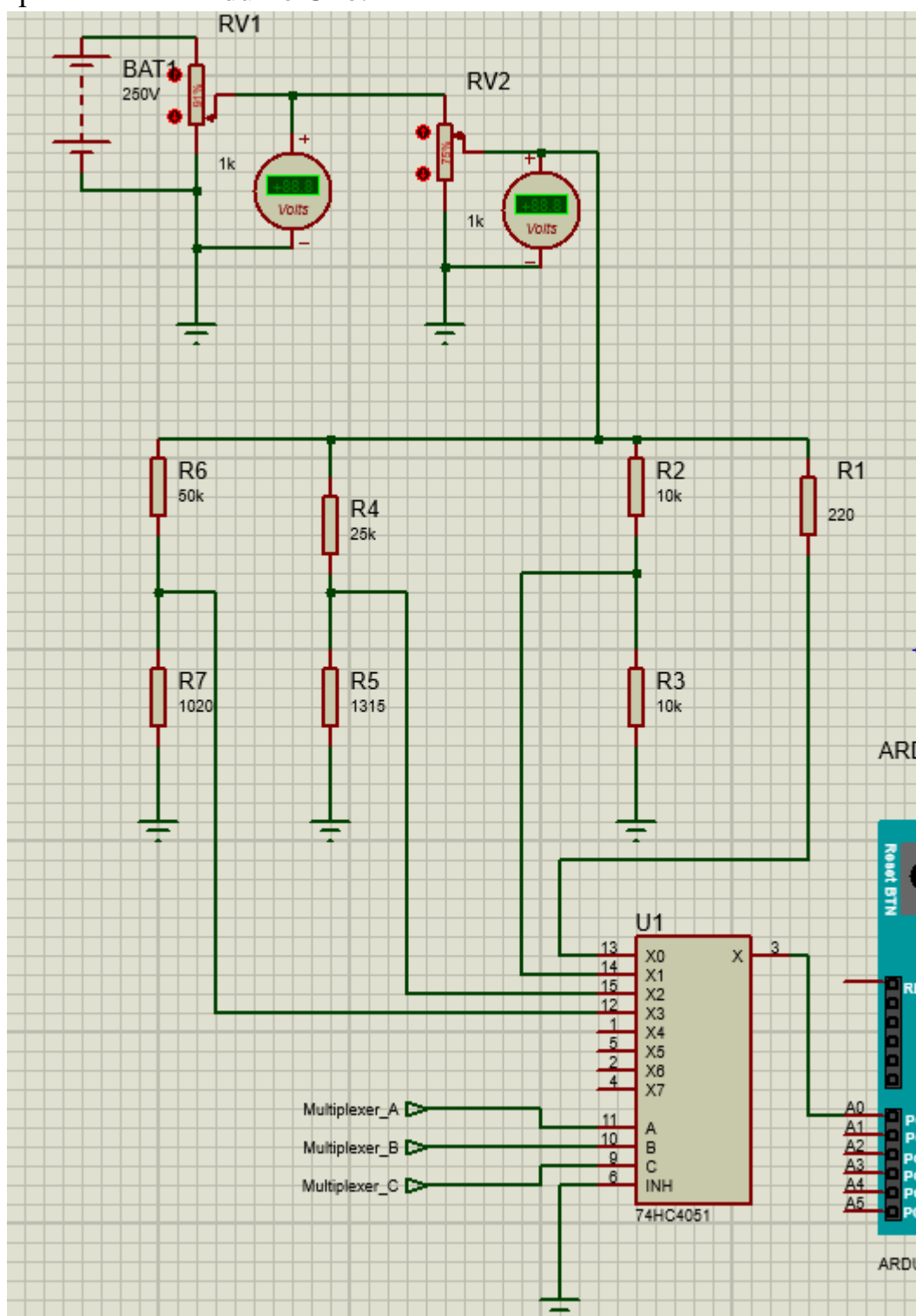


Рис 8. Схема подключения делителей напряжения к мультиплексору

3.2. Построение концептуальной модели системы

Для изменения напряжения на измеряемой нагрузке будем использовать 2 потенциометра сопротивлением 1кОм. Один потенциометр подключим к контактам источника питания, второй к выходу первого.

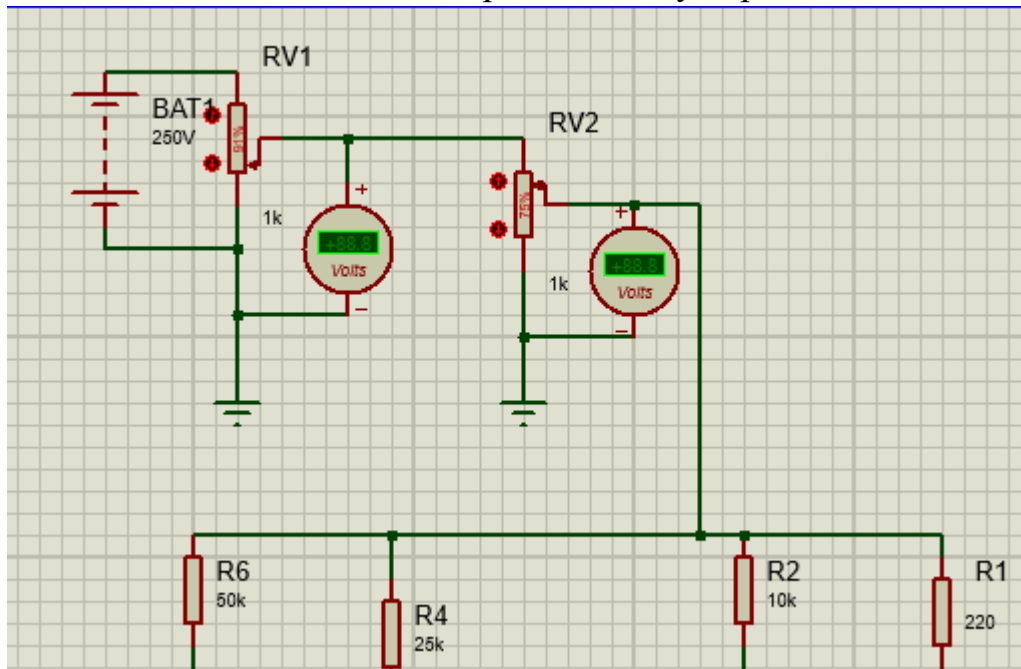


Рис 9. Схема подключения нагрузки

При напряжении источника питания равном 250В изменение сопротивления первого потенциометра на 1%, напряжение на его выходе будет меняться на 2.5 В. При установке нужного диапазона напряжения на первом потенциометре, с помощью второго можно установить более точное напряжение нагрузки с меньшим шагом.

Для проверки точности измерения разработанного вольтметра, подключим эталонные вольтметры к потенциометрам.

4. Разработка аппаратной части устройства

4.1. Выбор программного обеспечения для разработки

Для разработки и отладки аппаратной части устройства используется программа Proteus, которая представляет собой мощную систему схемотехнического моделирования, сделанной на основе виртуальных моделей электронных элементов. Специфической чертой данного программного пакета (Proteus) является возможность моделирования различной работы программируемых устройств: микропроцессоров, контроллеров, DSP и т.д.

Proteus прекрасно работает с большинством компиляторов и ассемблерами. PROTEUS VSM делает довольно достоверно моделирование и отладку весьма сложных устройств, в которых может находиться несколько микроконтроллеров разных семейств в одном устройстве. Необходимо учитывать и понимать, что любое моделирование электронных схем не может, абсолютно точно повторять работу реального устройства. Но для общей отладки, какого-либо алгоритма работы микропроцессорной системы, этого вполне будет достаточно. Программа PROTEUS имеет большую библиотеку электронных компонентов, а отсутствующие модели можно сделать самостоятельно.

4.2. Выбор элементов и проектирование схемы

Из библиотеки выберем необходимые элементы и добавим их в локальную библиотеку проекта для использования.

В качестве источника питания нагрузки выберем компонент BATTERY, в свойствах которого установим напряжение равное 250В.

Потенциометр представлен компонентом POT-HG, установим в его свойствах сопротивление равное 1кОМ.

Для делителей напряжения возьмем элемент RES, каждому резистору установим нужное сопротивление.

В качестве мультиплексора возьмем микросхему 74НС4051. Эта микросхема может работать в аналоговом режиме в качестве мультиплексора, или демультиплексора. Она имеет 8 каналов, по которым могут передаваться данные. Для разрабатываемой схемы будет использовано 4 канала.

Плата Arduino Uno содержится в библиотеке ArduinoUnoTEP.LIB, которую нужно установить отдельно.

4.3. Описание микроконтроллера

Плата Arduino Uno основана на микроконтроллере ATmega328P.

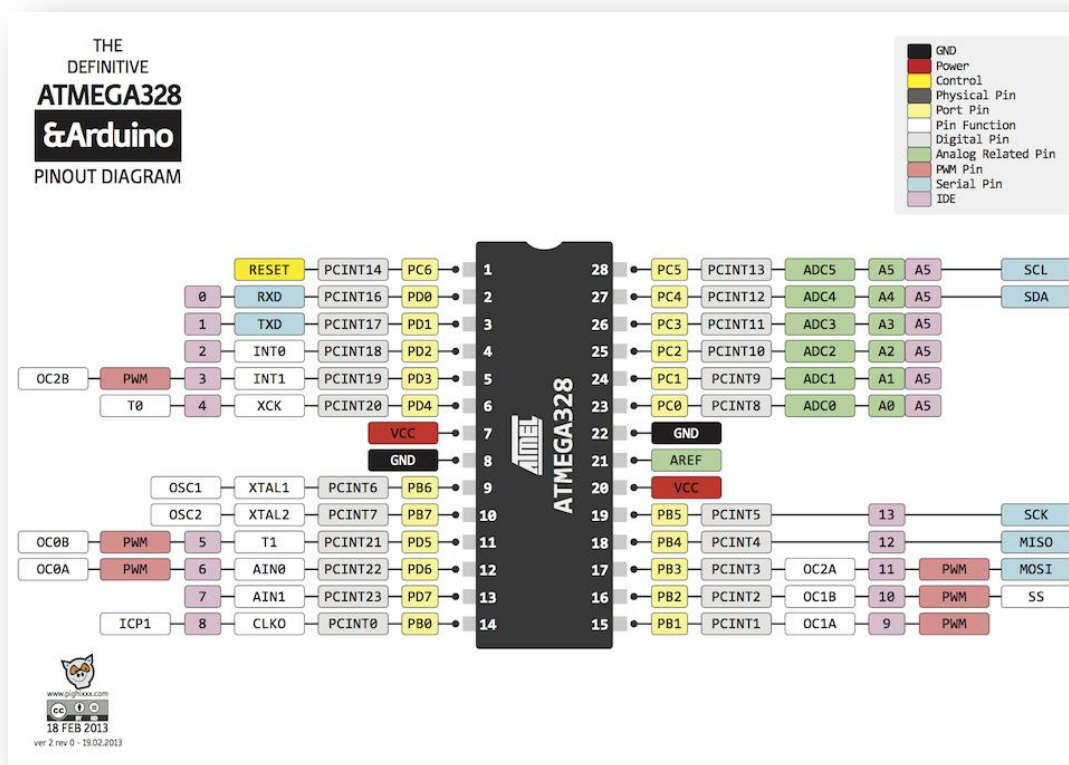


Рис 10. Конфигурация выводов Atmega 328P

Atmega328P - микроконтроллер семейства AVR, имеет 8-битный процессор и позволяет выполнять большинство команд за один такт.

Характеристики Atmega 328P:

Память:

- ❖ 32 kB Flash (память программ, имеющая возможность самопрограммирования)
- ❖ 2 kB ОЗУ
- ❖ 1 kB EEPROM (постоянная память данных)

Периферийные устройства:

- ❖ Два 8-битных таймера/счетчика с модулем сравнения и делителями частоты
- ❖ 16-битный таймер/счетчик с модулем сравнения и делителем частоты, а также с режимом записи
- ❖ Счетчик реального времени с отдельным генератором
- ❖ Шесть каналов PWM (аналог ЦАП)
- ❖ 6-канальный ЦАП со встроенным датчиком температуры
- ❖ Программируемый последовательный порт USART
- ❖ Последовательный интерфейс SPI
- ❖ Интерфейс I2C

- ❖ Программируемый сторожевой таймер с отдельным внутренним генератором
- ❖ Внутренняя схема сравнения напряжений
- ❖ Блок обработки прерываний и пробуждения при изменении напряжений на выводах микроконтроллера

Специальные функции микроконтроллера:

- ❖ Сброс при включении питания и программное распознавание снижения напряжения питания
- ❖ Внутренний калибруемый генератор тактовых импульсов
- ❖ Обработка внутренних и внешних прерываний
- ❖ 6 режимов сна (пониженное энергопотребление и снижение шумов для более точного преобразования АЦП)

Напряжения питания и скорость процессора:

- ❖ 1.8 - 5.5 В при частоте до 4 МГц
- ❖ 2.7 - 5.5 В при частоте до 10 МГц
- ❖ 4.5 - 5.5 В при частоте до 20 МГц

Исходя из приведённых выше характеристик, можно сделать вывод, что данный микроконтроллер обладает достаточной производительностью, количеством памяти и количеством портов ввода-вывода для решения поставленной задачи. Также, следует отметить, что под данный микроконтроллер существует несколько удобных сред разработки на высокоуровневых языках программирования, что упрощает процесс написания кода.

4.4. Принципиальная схема устройства и описание алгоритма работы

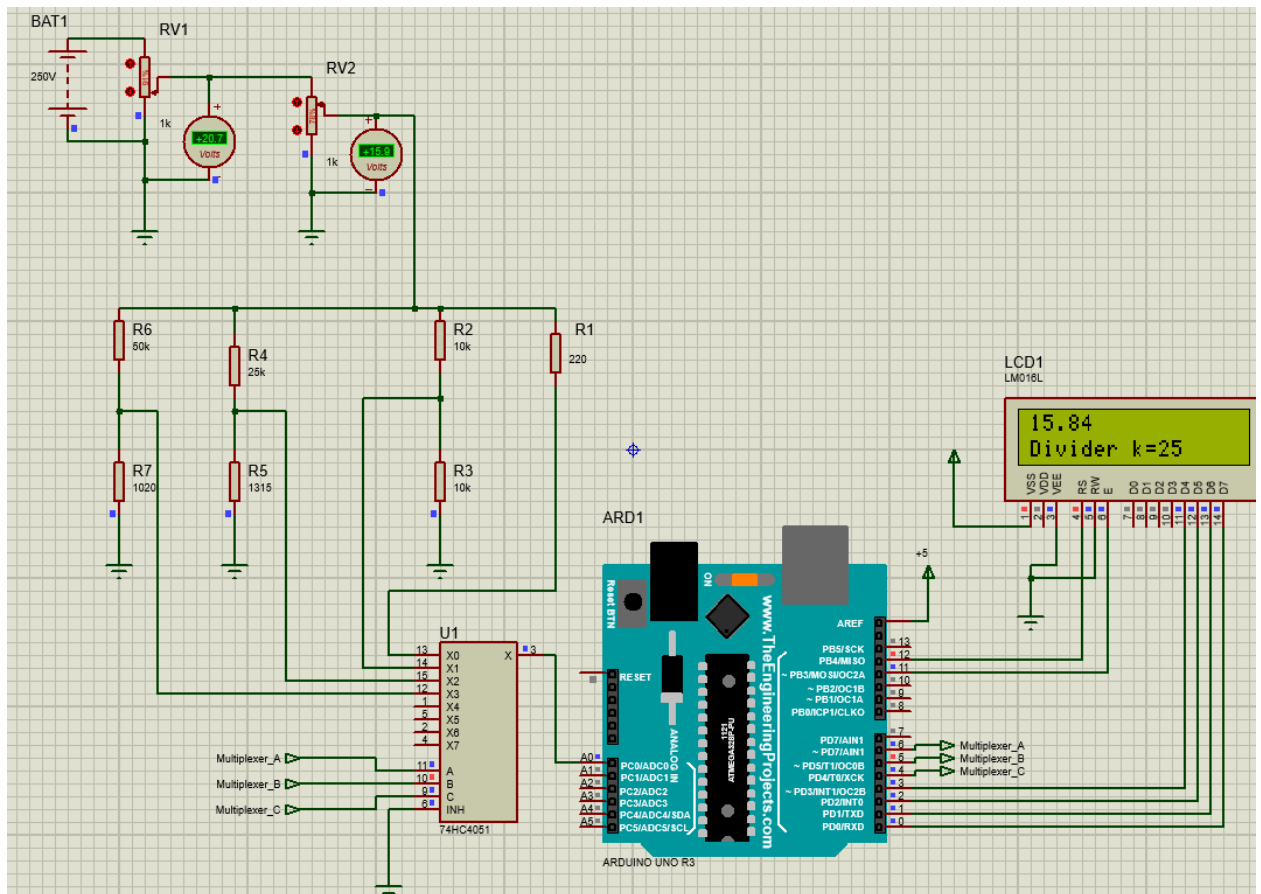


Рис 11. Принципиальная схема устройства

Напряжение со второго потенциометра подается на 4 параллельных линии – 3 линии являются делителя напряжения, на четвертой напряжение подается напрямую в мультиплексор.

В начальном состоянии микроконтроллер подает на управляющие (адресные) входы мультиплексора сигналы «1», «1», «0» - выбран третий адрес. По этому адресу находится делитель с самым большим коэффициентом – 50.

Выход мультиплексора подключен к АЦП платы. Поступающее напряжение пересчитывается в соответствии с коэффициентом делителя.

Если напряжение на нагрузке меньше 100В, то на мультиплексор от микроконтроллера передаются сигналы «0», «1», «0» - выбирается адрес 2. По этому адресу находится делитель с коэффициентом – 25. Напряжение поступает на АЦП и пересчитывается. Если оно меньше 10В, то принимаются данные с делителя с коэффициентом 2. Данные с последнего делителя пересчитываются в напряжение на нагрузке, если оно меньше 5, то мультиплексор передает данные с 0 адреса.

Программа работает циклично. После анализа напряжения на мультиплексор подается кодовое слово «110» - выбрать третий адрес и цикл анализа начинается заново.

5. Разработка программной части устройства

5.1. Выбор среды разработки

Для разработки и отладки программной части устройства я выбрал Arduino IDE. Данная IDE позволяет писать простые программы на C++ для множества микроконтроллеров, в том числе для Atmega 328P.

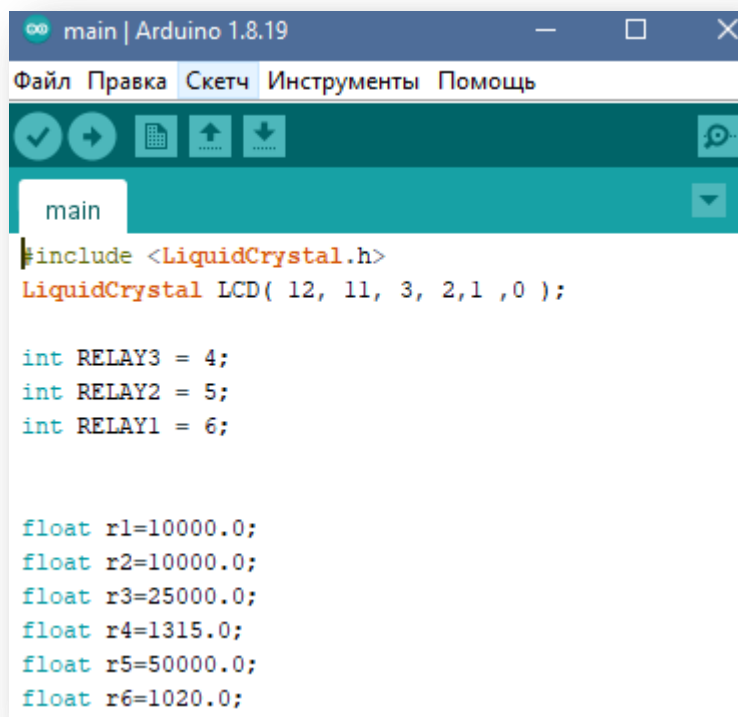
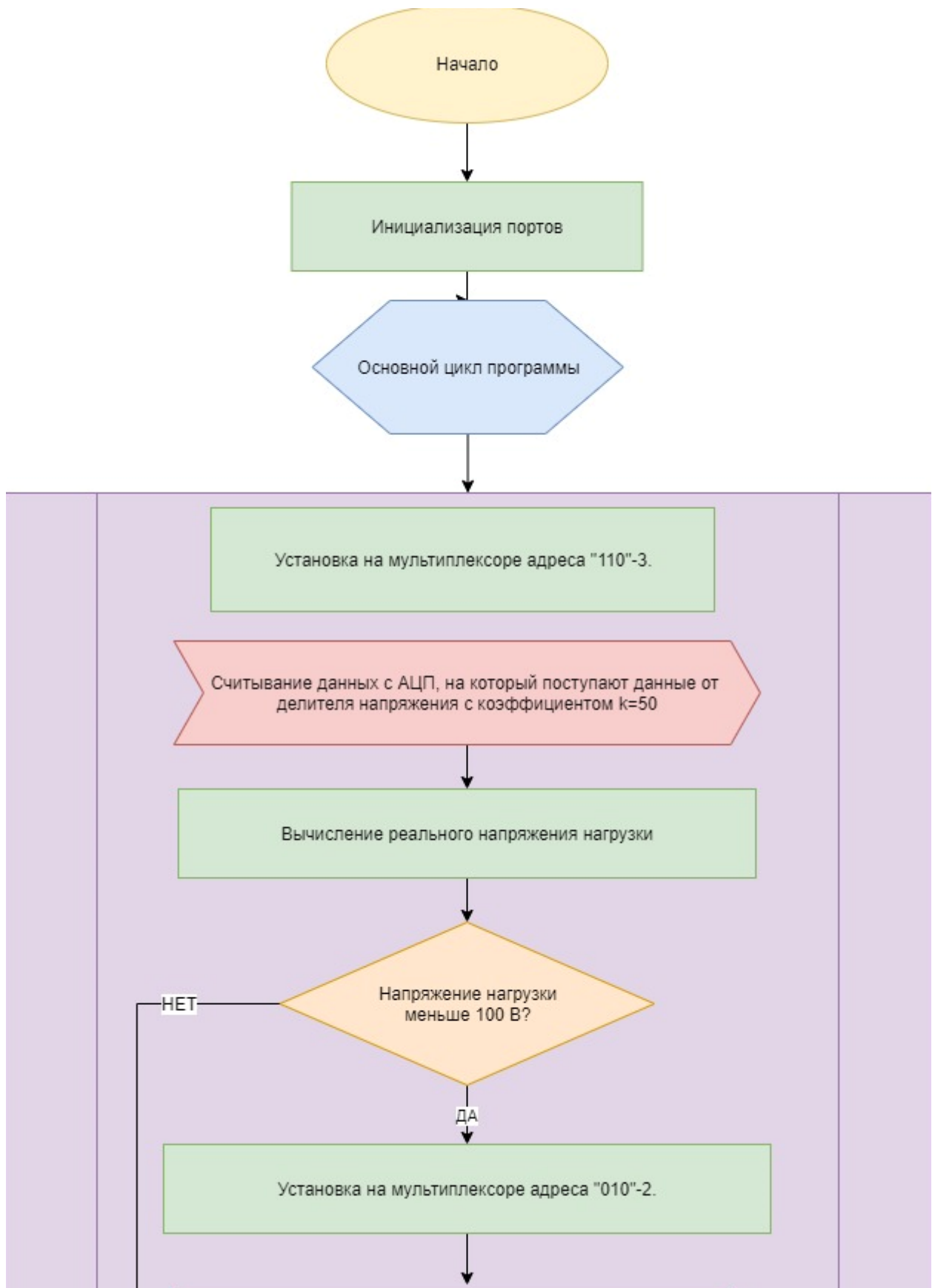
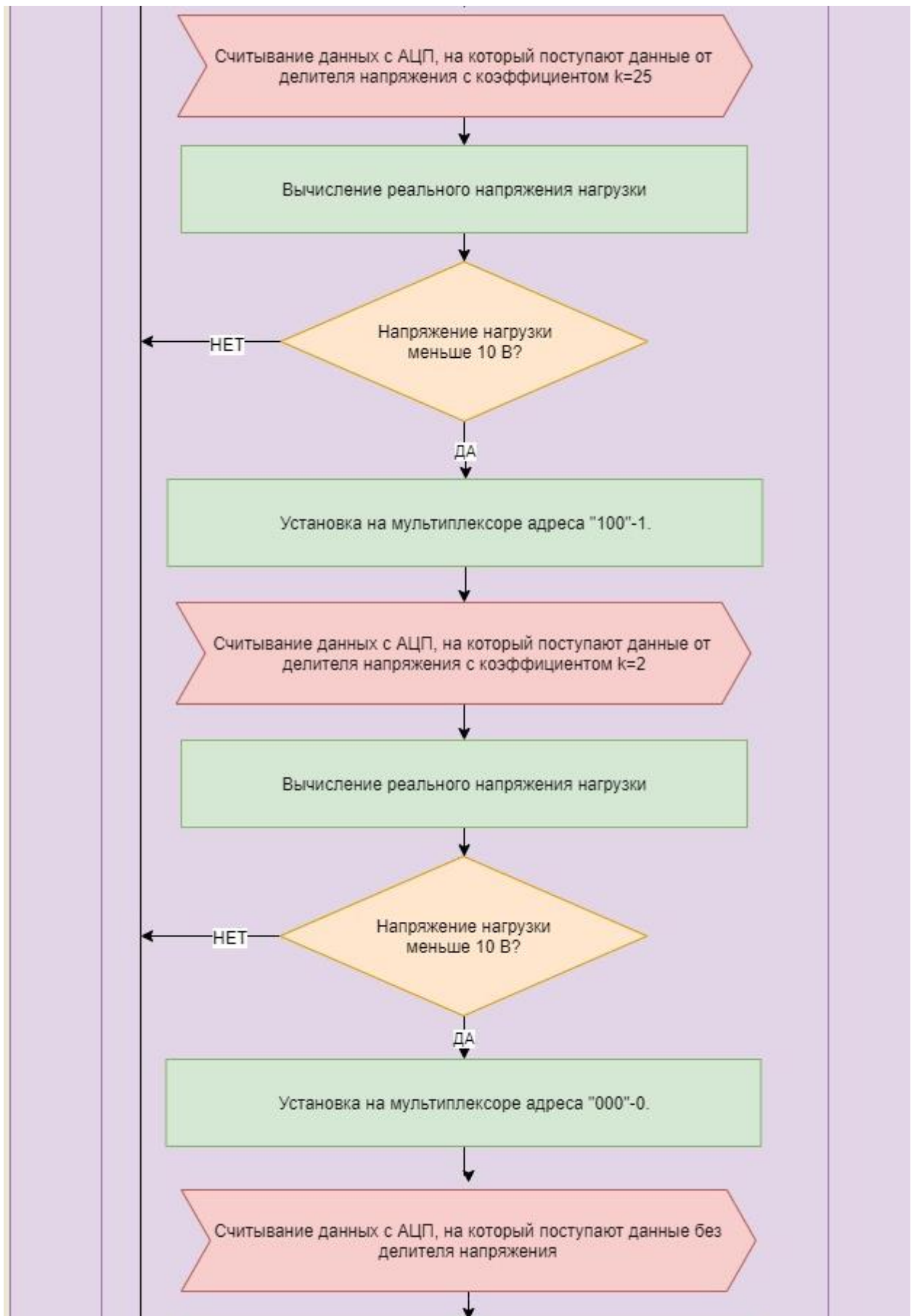
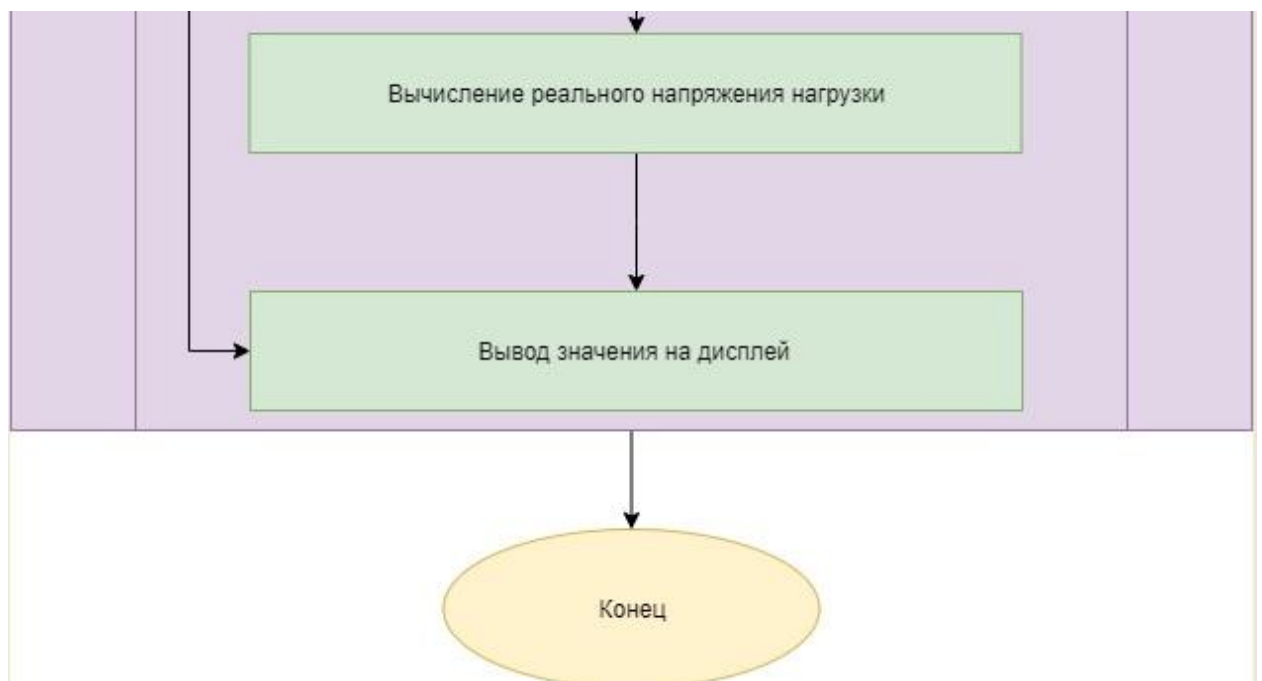


Рис 12. Arduino IDE

5.2. Алгоритм работы программы







6. Демонстрация работы устройства

6.1. Измерение напряжения в диапазоне от 0В до 250В

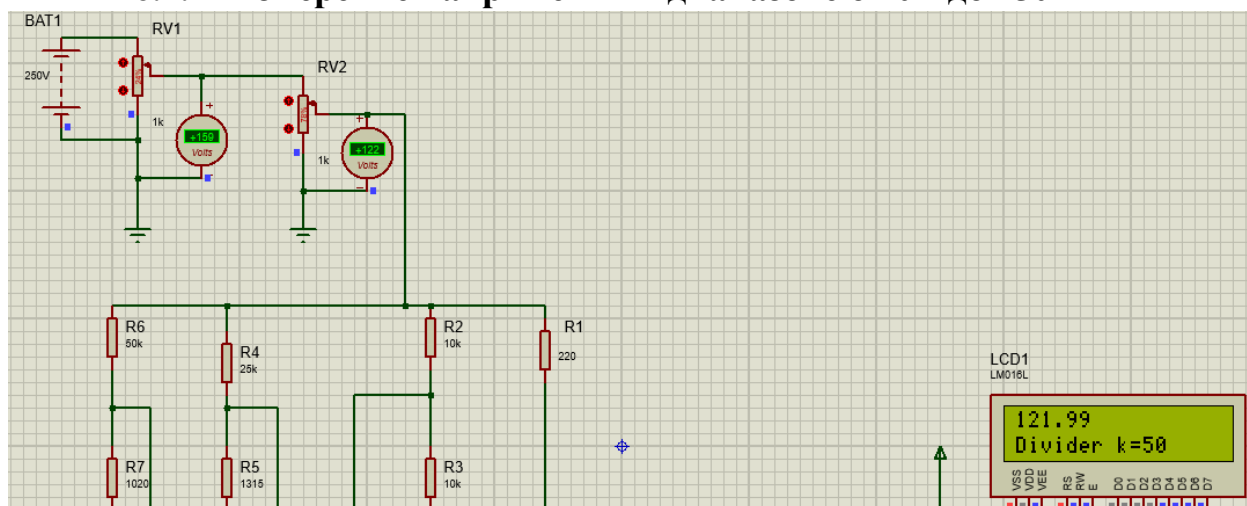


Рис 13.

Эталонный вольтметр на нагрузке определил напряжение равное 122В, разработанный вольтметр рассчитал напряжение, равное 121,99В.

6.2. Измерение напряжения в диапазоне от 0В до 100В

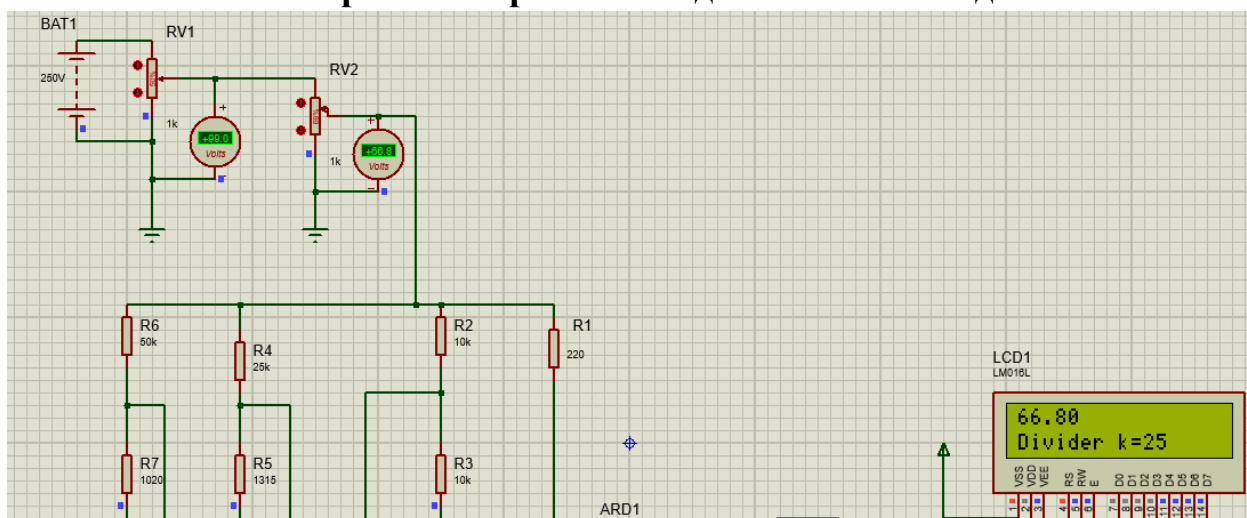


Рис 14.

Эталонный вольтметр на нагрузке определил напряжение равное 66,8В, разработанный вольтметр рассчитал напряжение, равное 66,80В.

6.3. Измерение напряжения в диапазоне от 0В до 10В

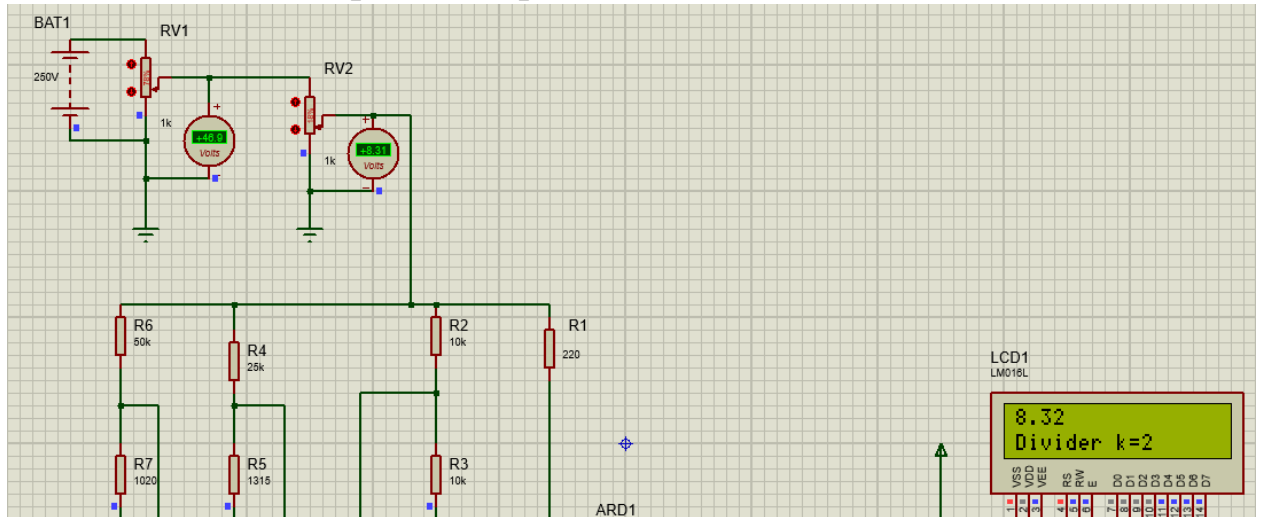


Рис 15.

Эталонный вольтметр на нагрузке определил напряжение равное 8,31В, разработанный вольтметр рассчитал напряжение, равное 8,32В.

6.4. Измерение напряжения в диапазоне от 0В до 5В

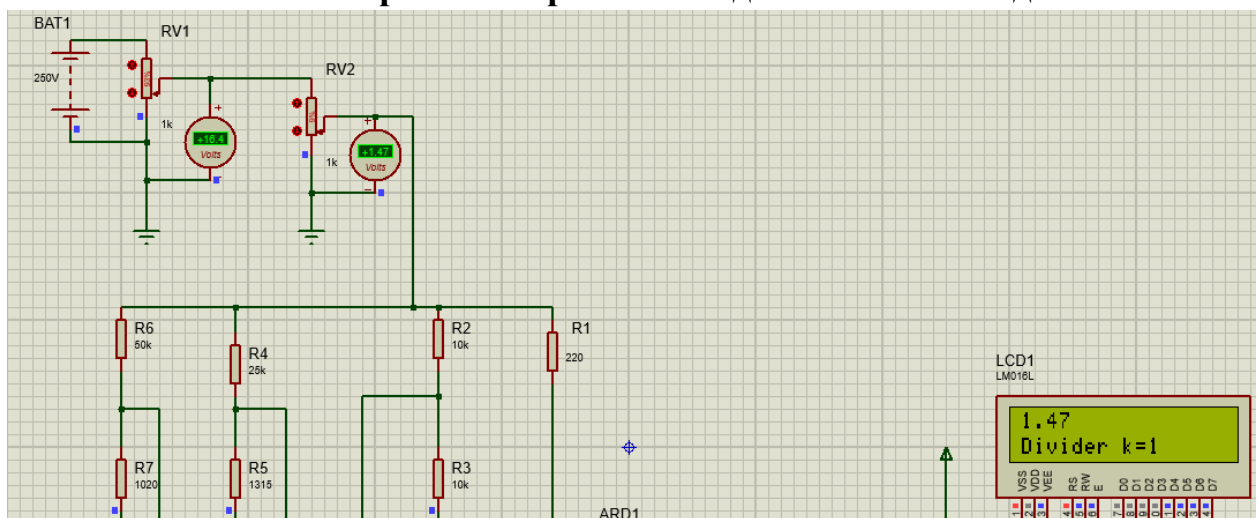


Рис 16.

Эталонный вольтметр на нагрузке определил напряжение равное 1,47В, разработанный вольтметр рассчитал напряжение, равное 1,47В.

После проведенных измерений можно сделать вывод о том, что разработанный вольтметр измеряет напряжение с заданной точностью на всех диапазонах.

7. Заключение.

В ходе выполнения курсовой работы были получены знания об устройстве и принципе работы цифрового вольтметра. Был предложен и спроектирован вариант схемы ЦВ, и проведено тестирование точности измерения.

8. Приложение 1. Исходный текст программы (листинг).

```
#include <LiquidCrystal.h>

LiquidCrystal LCD( 12, 11, 3, 2,1 ,0 );


int Multiplexer_A = 6;
int Multiplexer_B = 5;
int Multiplexer_C = 4;


float r2=10000.0;
float r3=10000.0;
float r4=25000.0;
float r5=1315.0;
float r6=50000.0;
float r7=1020.0;


float tv,rv;
int k;
float read_A0_adc(){
    int adc_voltage=analogRead(A0);
    float tmp_voltage=(adc_voltage/1023.0)*5.0;
    return tmp_voltage;
}
float real_voltage_calculation(int chanel,float tmp_voltage){
    float R_f,R_s,real_voltage;
    if(chanel==3){R_f=r6; R_s=r7;}
    if(chanel==2){R_f=r4; R_s=r5;}
    if(chanel==1){R_f=r2; R_s=r3;}
    real_voltage=tmp_voltage*(R_f+R_s)/R_s;
    return real_voltage;
}
```

```

void setup() {
    LCD.begin(16,2);
    pinMode(A0, INPUT);
    pinMode(Multiplexer_A, OUTPUT);
    pinMode(Multiplexer_B, OUTPUT);
    pinMode(Multiplexer_C, OUTPUT);
}

void loop() {
    digitalWrite(Multiplexer_A, HIGH);
    digitalWrite(Multiplexer_B, HIGH);
    digitalWrite(Multiplexer_C, LOW);

    tv=read_A0_adc();
    rv=real_voltage_calculation(3,tv);
    k=50;

    if(rv<100){
        digitalWrite(Multiplexer_A, LOW);
        tv=read_A0_adc();
        rv=real_voltage_calculation(2,tv);
        k=25;
        if (rv<10){
            digitalWrite(Multiplexer_A, HIGH);
            digitalWrite(Multiplexer_B, LOW);
            tv=read_A0_adc();
            rv=real_voltage_calculation(1,tv);
            k=2;
            if(rv<5){
                digitalWrite(Multiplexer_A, LOW);
                tv=read_A0_adc();
                rv=tv;
                k=1;
            }
        }
    }
}

```

```
        }  
    }  
}  
LCD.setCursor(0,0);  
LCD.print("          ");  
LCD.print(rv);  
LCD.setCursor(0,1);  
LCD.print("Divider k=");  
LCD.print(k);  
delay(300);  
}
```

9. Приложение 2. Информационные источники.

- ❖ <https://lemzspb.ru/printsip-raboty-tsifrovogo-vol-tmetra-postoyannogo-toka/>
- ❖ https://studopedia.ru/10_134478_tsifrovie-voltmetri.html
- ❖ <https://microkontroller.ru/arduino-projects>
- ❖ <https://www.ruselectronic.com/atmega328p/>