ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО

ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МЭИ» в г. СМОЛЕНСКЕ

Кафедра: вычислительной техники

Направление: 09.03.01 «Информатика и вычислительная техника»

Профиль подготовки: «Программное обеспечение»

Курсовая работа

по дисциплине: «Схемотехника»

Тема**:** «Секундомер на основе светодиодной матрицы**»**

Студент \_\_\_\_ПО1-19\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_Милославский С. А.\_

группа подпись фамилия И.О.

Руководитель \_\_к. т. н.\_\_\_\_\_ \_\_доцент\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_Аверченков О. Е.\_\_

учен. степень должность подпись фамилия И.О.

Смоленск, 2021 г.

АННОТАЦИЯ

Пояснительная записка содержит 150 листов печатного текста, 24 рисунков, 13 таблиц, список литературы из 20 наименований и 3 формулы.

Данный курсовой проект состоит из введения, двух разделов: проектирование и разработки курсовой работы, заключения, списка используемых источников и трех приложений.

В первом разделе курсовой работы выполняется проектирование курсовой работы, а именно анализ технического задания, описание структурной схемы, временные диаграммы и вывод основных формул.

Во втором разделе осуществляется разработка курсовой работы, а именно описание принципиальной схемы и используемых алгоритмов, оценка нормируемых параметров.

Результатом проекта является разработанное устройство – секундомер на основе светодиодной матрицы. Устройство позволяет подсчитывать время с точностью до 1/100 секунды. Оно способно отображать сотые доли секунды, секунды, минуты или часы в зависимости от режимов показа, который можно переключать с помощью кнопки. Также секундомер можно запускать, останавливать и сбрасывать.

Ключевые слова: секундомер, цифровой секундомер, x51, ОВМ, микроконтроллер, AT89C2051.

СОДЕРЖАНИЕ

[введение 4](#_Toc89079872)

[1. ПРоектирование курсовой работы 5](#_Toc89079873)

[1.1 Анализ технического задания 5](#_Toc89079874)

[1.2 Выбор и обоснование структурной схемы 5](#_Toc89079875)

[1.3 Описание принципа действия устройства по структурной схеме 5](#_Toc89079876)

[1.4 Временные диаграммы и вывод основных формул 5](#_Toc89079877)

[1.5 Выбор, обоснование и расчет отдельных элементов и узлов 5](#_Toc89079878)

[2. Реализация курсовой работы 6](#_Toc89079879)

[2.1 Схемы алгоритмов 6](#_Toc89079880)

[2.2 Оценка нормированных параметров 6](#_Toc89079881)

[2.2.1 Оценка общей потребляемой мощности 6](#_Toc89079882)

[2.2.2 Оценка потребляемого тока 6](#_Toc89079883)

[2.2.3 Оценка рабочих температур 6](#_Toc89079884)

[заключение 7](#_Toc89079885)

[список использованной литературы 8](#_Toc89079886)

[приложение А – перечень элементов 9](#_Toc89079887)

[Приложение Б – принципиальная схема 10](#_Toc89079888)

[приложение в – код программы 11](#_Toc89079889)

введение

Данная курсовая работа направлена на изучение проектирования устройств на базе ОВМ семейства x51. В рамках работы решается конкретная прикладная задача: разработка изделия, выполняющего функцию секундомера, способного подсчитывать время с точностью до 1/100 секунды.

Актуальность работы заключается в том, что секундомер является важным устройством с широкой областью применения. Он может использоваться в научных лабораториях, в заводских лаборатория, в учебных лабораториях ВУЗов, техникумов и школ, в спорте, в военном деле, а также в быту. Электронные секундомеры отличаются от механических повышенной точностью (1/100 сек) и более широким функционалом. Значимость работы подтверждается также и тем, что на рынке присутствуют сотни различных электронных секундомеров, цена на которые зачастую неоправданно завышена. В Интернете же возможно найти большое количество схем секундомеров на основе цифробуквенных индикаторов, но не на основе светодиодной матрицы, поэтому разработка дешевого устройства, в котором дисплей будет заменен на небольшую светодиодную матрицу. В качестве микроконтроллера будет взята недорогая микросхема их семейства x51.

Цель курсовой работы – разработка секундомера на основе светодиодной матрицы.

Для достижения поставленной цели были поставлены следующие задачи:

Изучить, используя литературу и Интернет, схемы других секундомеров и сделать на них обзор.

1. Нарисовать структурную схему и описать ее работу;
2. На основе структурной схемы нарисовать принципиальную схему;
3. Разработать общий алгоритм функционирования системы в словесном виде;
4. На основе составленного алгоритма написать программу на языке программирования C;
5. Откомпилировать программу и проверить ее работоспособность;
6. В случае наличия ошибок внести необходимые изменения, получив тем самым рабочий вариант программы.

Объектом исследования является проектирование устройств на базе ОВМ семейства x51.

Предметом исследования является разработка секундомера на основе светодиодной матрицы.

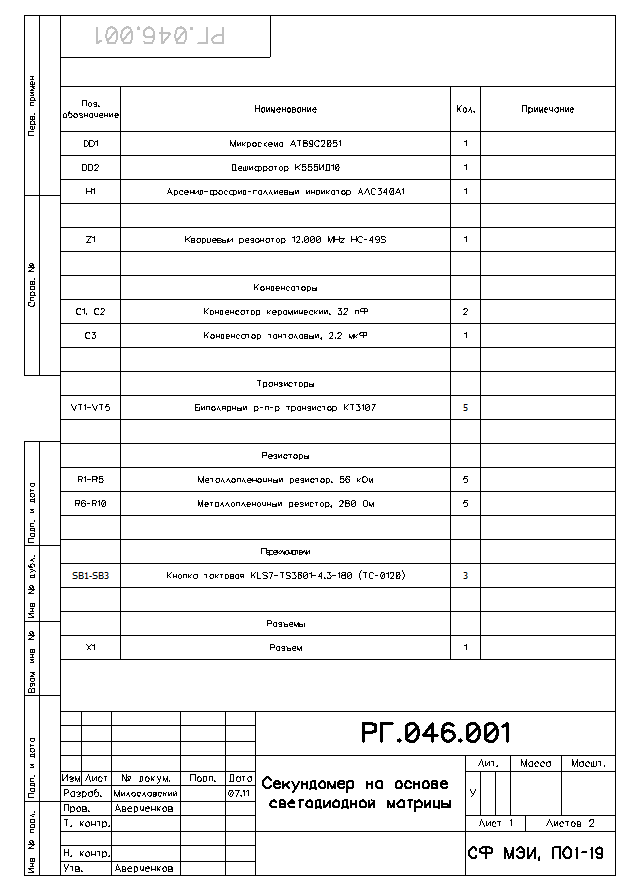
1. ПРоектирование курсовой работы
   1. Анализ технического задания
   2. Выбор и обоснование структурной схемы
   3. Описание принципа действия устройства по структурной схеме
   4. Временные диаграммы и вывод основных формул
   5. Выбор, обоснование и расчет отдельных элементов и узлов
2. Реализация курсовой работы
   1. Схемы алгоритмов
   2. Оценка нормированных параметров
      1. Оценка общей потребляемой мощности
      2. Оценка потребляемого тока
      3. Оценка рабочих температур

заключение

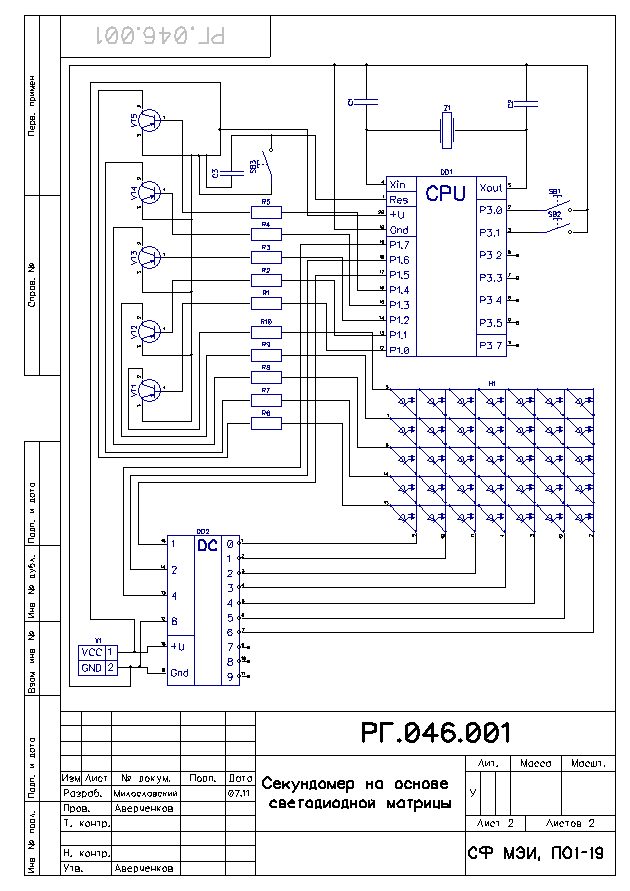
список использованной литературы

1. АВЕРЧЕНКОВ О.Е. Схемотехника: аппаратура и программы. ДМК Пресс, 2012. – 588 с.
2. АВЕРЧЕНКОВ О.Е. Сборник лабораторных работ по курсу «Схемотехника». СФ МЭИ, 2013.
3. АВЕРЧЕНКОВ О.Е. Сборник лабораторных работ по курсу «Микропроцессорные системы». СФ МЭИ, 2014.
4. Прошин А. А., Бростилов С. А., Горячев Н. В. Разработка цифрового секундомера // Молодой ученый. – 2015. – №2. – С. 187-190. – URL https://moluch.ru/archive/82/15066/ (дата обращения: 29.11.2021).
5. Курсовая работа: Электронный секундомер URL https://www.bestreferat.ru/referat-248590.html (дата обращения: 29.11.2021).
6. Простые часы на светодиодных матрицах. URL https://www.radiokot.ru/circuit/digital/home/103/ (дата обращения: 14.11.2021).

приложение А – перечень элементов



Приложение Б – принципиальная схема



приложение в – код программы

//Подключаем библиотеку

#include<at89x51.h>

//Тик таймера

#define tik (10000)

//Число столбцов в матрице

#define countColumn 7

//Ширина цифры

#define digitWidth 3

//Пауза 1мс

#define delay1ms delayFOR(99)

//Запущен или нет

unsigned char started = 0;

//Режим работы

//0 - показывать 1/100 секунды

//1 - показывать секунды

//2 - показывать минуты

//3 - показывать часы

unsigned char mode = 1;

//Указатель на столбец

unsigned char ptrColumn = 0;

//Время в 1/100 секунды

unsigned char time10ms = 0;

//Время в секундах

unsigned char timeSec = 0;

//Время в минутах

unsigned char timeMin = 0;

//Время в часах

unsigned char timeHour = 0;

//Массивы для хранения цифр

data unsigned char first[digitWidth];

data unsigned char second[digitWidth];

//Функция записи цифры в массив

void setData(unsigned char digit, unsigned char data \*array)

{

    switch(digit)

    {

        //(\*): если цира 0, то

        case 0:

            //В первый столбец записываем 1 + 2 + 4 + 8 + 16 = 1F = \*\*\*\*\*

            array[0] = 0x00;

            //Во второй столбец записываем 1 + 16 = 11 = \*---\*

            array[1] = 0x0E;

            //В третий столбец записываем 1 + 2 + 4 + 8 + 16 = 1F = \*\*\*\*\*

            array[2] = 0x00;

            //Итого получаем \*\*\*\*\*

            //               \*---\*

            //               \*\*\*\*\*

            //Повернуть по часовой стрелке; "\*" - светодиод горит, "-" - светодиод не горит

            break;

        case 1:

            array[0] = 0x00;

            array[1] = 0x1D;

            array[2] = 0x1B;

            break;

        case 2:

            array[0] = 0x08;

            array[1] = 0x0A;

            array[2] = 0x04;

            break;

        case 3:

            array[0] = 0x00;

            array[1] = 0x0A;

            array[2] = 0x0A;

            break;

        case 4:

            array[0] = 0x00;

            array[1] = 0x1B;

            array[2] = 0x18;

            break;

        case 5:

            array[0] = 0x02;

            array[1] = 0x0A;

            array[2] = 0x08;

            break;

        case 6:

            array[0] = 0x02;

            array[1] = 0x0A;

            array[2] = 0x00;

            break;

        case 7:

            array[0] = 0x1C;

            array[1] = 0x1A;

            array[2] = 0x06;

            break;

        case 8:

            array[0] = 0x00;

            array[1] = 0x0A;

            array[2] = 0x00;

            break;

        case 9:

            array[0] = 0x00;

            array[1] = 0x0A;

            array[2] = 0x08;

            break;

    }

}

//Функция переключения столбца

void nextColumn(void)

{

    //Последние 3 бита регистра P1 отвечают за то, какой столбец показывать

    //Они подаются на дешифратор, который выдает значения из множества {0,1,2,3,4,5,6} (Последние три ножки дешифратора не используются)

    //Исходя из сказанного, в P1 нужно записать указатель на столбец смещенный на 5 битов влево побитно сложенный со значением из массива, хранящего цифру, по указателю

    if (ptrColumn < 3)

        P1 = (ptrColumn << 5) | second[ptrColumn];

    //Если промежуток, то ничего не выводим

    if (ptrColumn == 3)

        P1 = (ptrColumn << 5) | 0x1F;

    //Если 2 цифра, то берем из массива second

    if (ptrColumn > 3 && ptrColumn < 7)

        P1 = (ptrColumn << 5) | first[ptrColumn - 4];

    //Переходим к следующему столбцу

    ptrColumn++;

    //Если столбец был последним, то переходим опять к первому

    if (ptrColumn == countColumn)

        ptrColumn = 0;

}

//Функция записи цифр в зависимости от режима

//Работает аналогично примеру из книги

void setDigits()

{

    switch(mode)

    {

        case 0:

            setData(time10ms/10, &first[0]);

            setData(time10ms%10, &second[0]);

            break;

        case 1:

            setData(timeSec/10, &first[0]);

            setData(timeSec%10, &second[0]);

            break;

        case 2:

            setData(timeMin/10, &first[0]);

            setData(timeMin%10, &second[0]);

            break;

        case 3:

            setData(timeHour/10, &first[0]);

            setData(timeHour%10, &second[0]);

            break;

    }

}

//Инициализация таймера

void T0init(void)

{

    TR0=0;

    TMOD=(TMOD&0xf0)|0x1;

    TL0=(~tik);

    TH0=(~tik)>>8;

    TR0=1;

    ET0=1;

    EA=1;

}

//Прерывание таймера;

void T0\_int (void) \_\_interrupt (TF0\_VECTOR)

{

    TR0=0;

    TL0=(~tik);

    TH0=(~tik)>>8;

    TR0=1;

    if (started)

    {

        time10ms++;

        if (time10ms == 100)

        {

            time10ms = 0;

            timeSec++;

            if (timeSec == 60)

            {

                timeSec = 0;

                timeMin++;

                if (timeMin == 60)

                {

                    timeMin = 0;

                    timeHour++;

                    if (timeHour == 24)

                        timeHour = 0;

                }

            }

        }

        setDigits();

    }

}

//Пауза с помощью FOR

void delayFOR(unsigned char kodF)

{

    unsigned char ml;

    for(ml = 0; ml < kodF; ml++){}

}

//Пауза с заданным числом мс

void DelayMS(unsigned char KodMS)

{

    do delay1ms;

    while(--KodMS);

}

//Обработка нажатия кнопок

void buttonsCheck(void)

{

    //Пауза и старт секундомера

    if (!P3\_2)

    {

        //Выключаем матрицу

        P1 = 0x1F;

        //Игнорируем дребезг

        DelayMS(10);

        //Ждем отпускания

        while(!P3\_2);

        //Игнорируем дребезг

        DelayMS(10);

        //Запускаем или останавливаем секундомер

        if (started)

        {

            started = 0;

        } else

        {

            started = 1;

        }

    }

    //Переключение режимов

    if (!P3\_3)

    {

        //Выключаем матрицу

        P1 = 0x1F;

        //Игнорируем дребезг

        DelayMS(10);

        //Ждем отпускания

        while(!P3\_3);

        //Игнорируем дребезг

        DelayMS(10);

        //Меняем режим

        mode++;

        if (mode == 4)

            mode = 0;

        //Устанавливаем правильные цифры в зависимости от режима

        setDigits();

    }

    //Сброс секундомера

    if (!P3\_4)

    {

        //Сброс значений

        DelayMS(10);

        time10ms=0;

        //Время в секундах

        timeSec = 0;

        //Время в минутах

        timeMin = 0;

        //Время в часах

        timeHour = 0;

        DelayMS(10);

        setDigits();

    }

}

void main(void){

    //Инициализируем таймер

    T0init();

    //Устанавливаем цивры (0, 0)

       setDigits();

    while(1){

        //Проверяем не нажал ли пользователь на кнопку

        buttonsCheck();

        //Делаем паузу

        delayFOR(10);

        //Меняем столбец

        nextColumn();

    }

}