МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра ЭВМ

Отчёт

Лабораторная работа № 4 по дисциплине

«Микропроцессорные системы»

Выполнили студенты группы ИВТб-4301 \_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Колесников Р. К./

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Крючков И. С./

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Бушков Д. А./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Крутиков А. К./

Киров 2023

1. Задание

1) Разработать пешеходный светофор со следующими состояниями: красный, мигающий красный, зеленый, мигающий зеленый.

2) Разработать простой калькулятор со следующими операциями: «+», «-», «/», «\*».

1. Структурная и принципиальная схема лабораторной установки

Структурная схема лабораторной установки представлена на рисунке 1.

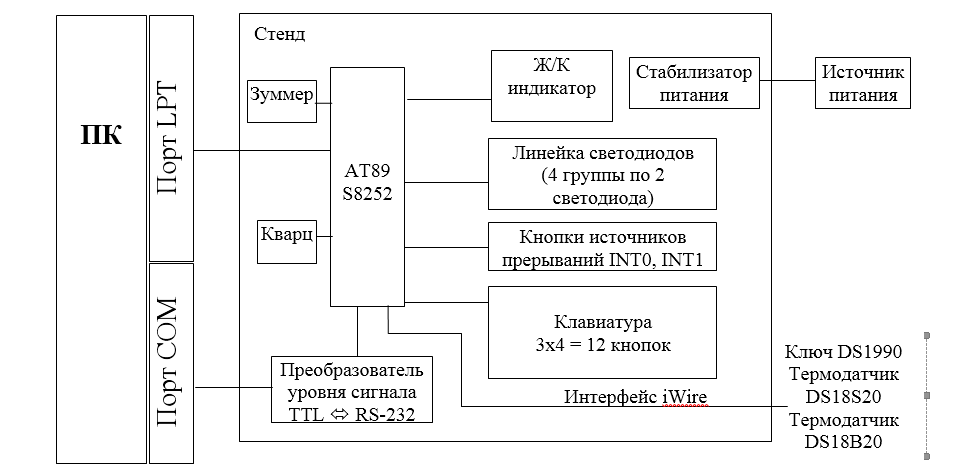


Рисунок 1 – Структурная схема лабораторной установки

Принципиальная схема лабораторной установки представлена на рисунке 2.

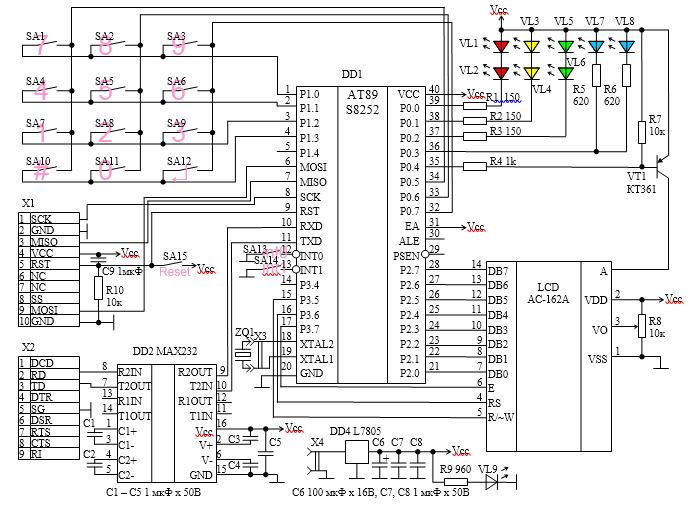


Рисунок 2 – Принципиальная схема лабораторной установки

1. Ход работы
   1. Задание 1. Светофор

Светофор представляет из себя два светодиода – красный и зеленый. Сначала загорается красный светодиод, по истечению некоторого промежутка времени красный светодиод начинает мигать, после чего загорается зелёный. При переходе с зеленого на красный сигнал также происходит мигание зеленого светодиода.

Модель управляющего автомата для светофора представлена на рисунке 3. Код программы представлен в приложении А.

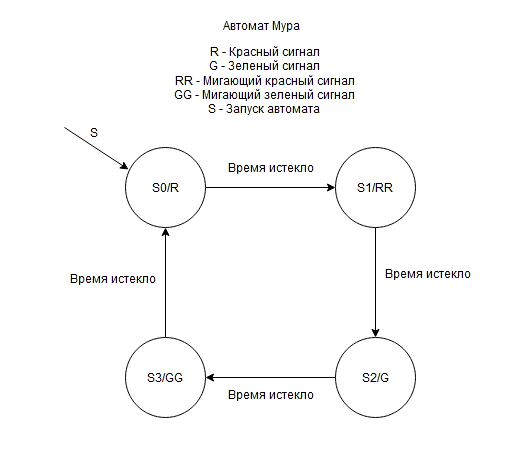


Рисунок 3 – Модель управляющего автомата для светофора

* 1. Задание 2. Калькулятор

Калькулятор представляет из себя клавиатуру и дисплей. Сначала пользователю предоставляется ввести число, выбрать арифметическую операцию из доступных, ввести второе число, после чего на дисплей будет выведен результат.

Модель управляющего автомата для калькулятора представлена на рисунке 4. Код программы представлен в приложении Б.

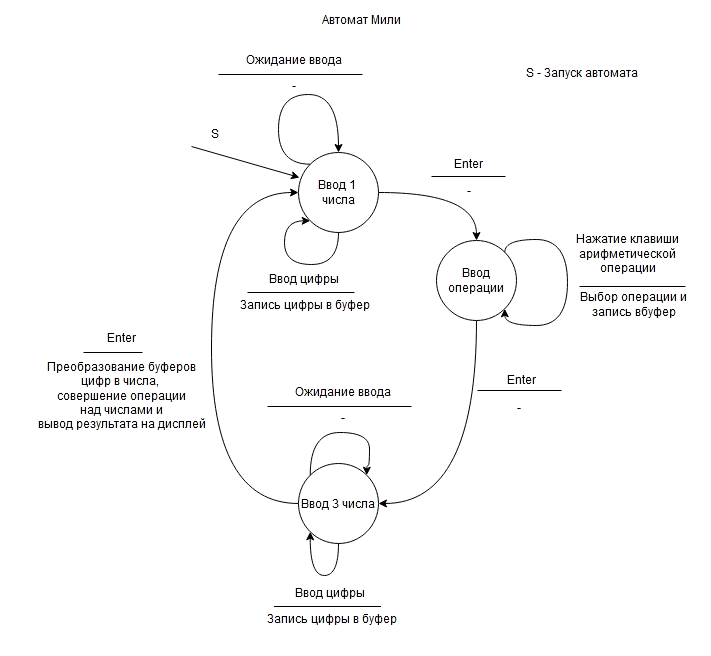


Рисунок 4 – Модель управляющего автомата для калькулятора

1. Выводы

В ходе лабораторной работы на примере микроконтроллера АТ89S8252 были изучены вопросы проектирования микроконтроллерных систем и получены знания и навыки, необходимые для проектирования таких систем.

Микроконтроллер AT89S8252 представляет собой расположенные на одном кристалле CISC-процессор, оперативную память, энергонезависимые память данных и память программ, средства обработки прерываний (шесть векторов, девять источников), три 16-разрядных таймера/счетчика, сторожевой таймер, интерфейсы SPI и UART, четыре восьмиразрядных квазидвунаправленных порта ввода-вывода, тактовый генератор. Контроллер является совместимым (по выводам и программно) с микроконтроллерами серий C51, C52. При этом по отношению к контроллерам C51 имеются дополнительные функциональные возможности.

Процессор выполняет команды, коды которых записаны в энергонезависимую память программ (Flash-память). На выполнение одной команды в большинстве случаев требуется один или два машинных цикла. Машинный цикл состоит из двенадцати тактов. Таким образом, время выполнения каждой команды может быть рассчитано по известной частоте тактового генератора. Времязадающим элементом тактового генератора служит кварцевый резонатор, для которого на плате стенда предусмотрен разъем.

Встроенный контроллер прерываний позволяет обрабатывать прерывания от девяти источников, распределяя их между шестью векторами: два вектора прерывания от внешних источников (предусмотрены две кнопки на плате стенда) – как по фронту, так и по уровню сигнала, три вектора прерывания от встроенных таймеров, вектор прерывания от последовательных интерфейсов. Прерывания обслуживаются по дисциплине фиксированных приоритетов. При этом для каждого вектора можно установить один из двух приоритетов.

Таймеры-счетчики могут быть использованы в качестве генератора тактовой частоты для интерфейса UART, счета времени, счета внешних событий, измерения длительности сигнала, частоты и скважности периодических сигналов, генератора частоты. Поскольку 16-разрядные таймеры/счетчики в качестве исходной используют тактовую частоту, период счета зависит от частоты выбранного кварцевого резонатора.

Были написаны две программы для микроконтроллера АТ89S8252 на языке С: пешеходный светофор и калькулятор.

Приложение А

(Обязательное)

Листинг программы светофора

//#include "89xs8252.h"

//#include "temperature.h"

#include "drv.h"

#include "1wire.h"

#define R 1 // Код состояния

#define RY 2

#define G 3

#define GG 4

#define Y 5

#define SR 6

#define SG 7

char key=0; // Идентификатор нажатой кнопки

char state=R; // Идентификатор управляющего состояния

char t; // Значение счетчика

char tr; // Время красного сигнала, с

char tg; // Время зеленого сигнала, с

char T\_FLAG = 0; // Сигнал счетчика "время истекло"

int ms = 0; // Счетчик милисекунд

void main (void) {

// Инициализация

init();

RED = 0; // Красный цвет

YEL = 1; // Желтый цвет

GRN = 1; // Зеленый цвет

lcd\_led = 0; // Подсветка индикатора

tr=rd\_EEPROM(0); if((tr>30)||(tr<5)) tr=5;

tg=rd\_EEPROM(1); if((tg>30)||(tg<5)) tg=5;

for(t=0;t!=0;t++); // тест задержки

t = 10\*tr;

// Реализация управляющего автомата

while (1) {

//key=ScanKbd();

switch (state) {

case R: // Красный

clear\_lcd(); outcw(0x80); outd('К');

if (T\_FLAG) { state=RY; T\_FLAG = 0; t=1; YEL=1; }

break;

case RY: // Красный-желтый

clear\_lcd(); outcw(0x80); outd('К'); outd('р'); outd('.'); outd('-');

outd('ж'); outd('.');

RED=0; DelayMs(500);

RED=1; DelayMs(400);

if (T\_FLAG) { state=G; T\_FLAG = 0; t=10; YEL=1; GRN=0;}

//if (T\_FLAG) { state=G; T\_FLAG = 0; t=10\*tg; RED=1; YEL=1; GRN=0; }

break;

case G: // Зеленый

clear\_lcd(); outcw(0x80); outd('З');

if (T\_FLAG) { state=GG; T\_FLAG = 0; t= 3; RED = 1;}

break;

case GG: // Мигающий зеленый

GRN=0; DelayMs(500);

GRN=1; DelayMs(400);

if (T\_FLAG) { state=R; T\_FLAG = 0; t=10; YEL=1;RED = 0;}

break;

}

DelayMs(100); // такт работы автомата

if(t==0) T\_FLAG=1; else t--; // счетчик

}

}

/\*

void DelayMs(unsigned int m){ // задержка пустым циклом

unsigned char a; //счетчик милисекундного цикла

//unsigned int ms; //счетчик милисекунд

for(ms=0;ms!=m;ms++) {

for(a=0;a!=120;a++); // пустой цикл 1мс

//WMCON.WDTRST=1; // сброс сторожевого таймера

}

}

\*/

void DelayMs(unsigned int m){ // задержка по таймеру

//unsigned char a;

//a = ms+(m<<1);

ms=0;

WMCON.WDTRST=1; // сброс сторожевого таймера

while(ms!=m) continue;

}

void Timer1InterruptHandler() org IVT\_ADDR\_ET1{

EA\_bit = 0; // Clear global interrupt enable flag

TF1\_bit = 0; // Ensure that Timer1 interrupt flag is cleared

TR1\_bit = 0; // Stop Timer1

TH1 = 0xFC; // Reset Timer1 high byte 65536-1000

TL1 = 0x18; // Reset Timer1 low byte

//P0 = ~P0; // Toggle PORT0

ms++;

EA\_bit = 1; // Set global interrupt enable flag

TR1\_bit = 1; // Run Timer1

}

void INT0\_Interrupt() org IVT\_ADDR\_EX0 {

EA\_bit = 0;

lcd\_led=~lcd\_led;

EA\_bit = 1;

}

/\*

void UART\_Interrupt() iv IVT\_ADDR\_ES {

if (RI\_bit){

if (SBUF==0x13) {state=SR; RED=1; YEL=1; GRN=1;}

RI\_bit=0;

}

if (TI\_bit) TI\_bit = 0;

}

\*/

Приложение Б

(Обязательное)

Листинг программы калькулятора

//#include "89xs8252.h"

//#include "temperature.h"

#include "drv.h"

#include "1wire.h"

#include "string.h"

#define INPUT\_FN 1 //Состояние ввода первого числа

#define SELECT\_OP 2 //Состояние выбора операции

#define INPUT\_SN 3 //Состояние ввода второго числа

unsigned char key; // Идентификатор нажатой кнопки

char state=INPUT\_FN; // Идентификатор управляющего состояния

char fN[4]; //первое число

char sN[4]; //второе число

char op = '/'; //арифм операция

char result[8]; //результат (строка)

int res = 0; //целочисленный результат

//вспомогательные переменные

int i, j;

int isNegative;

int length;

int ms = 0; // Счетчик милисекунд

//Процедура ввода добавления цифры в число

void inputNumber(char number[], char key)

{

int len = strlen(number);

if (key == '0' && len == 1) {clear\_lcd(); outcw(0x80); return;}

//if (len == 4) return;

number[len] = key;

number[len + 1] = '\0';

}

//конвертация целочисл результата в строку

void intToString(int number, char str[])

{

i = 0;

isNegative = 0;

//проверка на отрицательное число

if (number < 0) {

isNegative = 1;

number = -number;

}

//запись цифр в обратном порядке

while (number > 0) {

int digit = number % 10;

str[i++] = '0' + digit;

number /= 10;

}

//добавление минуса если число отрицательное

if (isNegative) {

str[i++] = '-';

}

str[i] = '\0';

//обратный порядок цифр в строке

length = i;

for (j = 0; j < length / 2; j++)

{

char temp = str[j];

str[j] = str[length - j - 1];

str[length - j - 1] = temp;

}

}

void main (void)

{

// Инициализация

init();

RED = 1; // Красный цвет

YEL = 1; // Желтый цвет

GRN = 0; // Зеленый цвет

lcd\_led = 0; // Подсветка индикатора

clear\_lcd();

outcw(0x80);

while (1)

{

key=ScanKbd(); //опрашиваем кнопку

switch (state)

{

case INPUT\_FN:

if (key == 13 && strlen(fN) != 0) {state = SELECT\_OP; clear\_lcd(); outcw(0x80); break;}

if (key == '#') break;

if (strlen(fN) == 4) break;

inputNumber(fN, key);

outd(key);

break;

case SELECT\_OP:

if (key == '#')

{

if (op == '/') {op = '+'; RED = 1; YEL = 1;}

else if (op == '+') {op = '-'; RED = 0; YEL = 1;}

else if (op == '-') {op = '\*'; RED = 1; YEL = 0;}

else if (op == '\*') {op = '/'; RED = 0; YEL = 0;}

}

if (key == 13) state = INPUT\_SN;

break;

case INPUT\_SN:

if (key == 13 && strlen(sN) != 0)

{

switch (op)

{

case '+':

res = atoi(fN) + atoi(sN);

break;

case '-':

res = atoi(fN) - atoi(sN);

break;

case '\*':

res = atoi(fN) \* atoi(sN);

break;

case '/':

res = atoi(fN) / atoi(sN);

break;

}

intToString(res, result);

outcw(0x88);

outd(result[0]); outd(result[1]);

outd(result[2]); outd(result[3]);

outd(result[4]); outd(result[5]);

outd(result[6]); outd(result[7]);

RED = 1; YEL = 1;

//очистка массивов

fN[0] = '\0'; sN[0] = '\0';

fN[1] = '\0'; sN[1] = '\0';

fN[2] = '\0'; sN[2] = '\0';

fN[3] = '\0'; sN[3] = '\0';

result[0] = '\0'; result[1] = '\0';

result[2] = '\0'; result[3] = '\0';

result[4] = '\0'; result[5] = '\0';

result[6] = '\0'; result[7] = '\0';

outcw(0x80);

state = INPUT\_FN;

break;

}

if (key == '#') break;

if (strlen(sN) == 4) break;

inputNumber(fN, key);

outd(key);

break;

}

DelayMs(100);

}

}

void DelayMs(unsigned int m){ // задержка по таймеру

//unsigned char a;

//a = ms+(m<<1);

ms=0;

WMCON.WDTRST=1; // сброс сторожевого таймера

while(ms!=m) continue;

}

void Timer1InterruptHandler() org IVT\_ADDR\_ET1{

EA\_bit = 0; // Clear global interrupt enable flag

TF1\_bit = 0; // Ensure that Timer1 interrupt flag is cleared

TR1\_bit = 0; // Stop Timer1

TH1 = 0xFC; // Reset Timer1 high byte 65536-1000

TL1 = 0x18; // Reset Timer1 low byte

//P0 = ~P0; // Toggle PORT0

ms++;

EA\_bit = 1; // Set global interrupt enable flag

TR1\_bit = 1; // Run Timer1

}

void INT0\_Interrupt() org IVT\_ADDR\_EX0 {

EA\_bit = 0;

lcd\_led=~lcd\_led;

EA\_bit = 1;

}

/\*

void UART\_Interrupt() iv IVT\_ADDR\_ES {

if (RI\_bit){

if (SBUF==0x13) {state=SR; RED=1; YEL=1; GRN=1;}

RI\_bit=0;

}

if (TI\_bit) TI\_bit = 0;

}

\*/