


Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

Работа выполнена в СКБ «Промышленная робототехника»

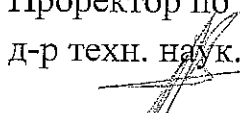
СОГЛАСОВАНО

Начальник отдела ОНИПКРС



(подпись) Е.М. Димитриади
« 19 » 06 2023 г.

УТВЕРЖДАЮ

Проректор по научной работе,
д-р техн. наук, профессор


(подпись) А.В. Космынин
« 19 » 06 2023 г.

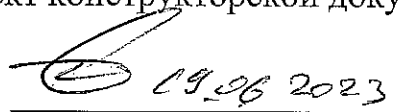
Декан ФЭУ


(подпись) А.С. Гудим
« 19 » 06 2023 г.

Аппаратно-программный комплекс
«Контроллер для управления силовой нагрузкой по расписанию»

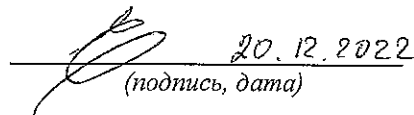
Комплект конструкторской документации

Руководитель СКБ


(подпись, дата)

С.И. Сухоруков

Руководитель проекта


(подпись, дата)

В.А. Егоров

Карточка проекта

Название	Контроллер для управления силовой нагрузкой по расписанию
Тип проекта	Тип проекта: учебная работа
Исполнители	Д. Д. Авакумов, Д. А. Логинов
Срок реализации	14.09.2022- 31.12.2022

Использованные материалы и компоненты

QTY	PART-REFS	VALUE	CODE
Resistors			
6	R1-R6	4.7K	Digikey 311-4.7KETR-ND
Integrated Circuits			
1	U1	ATMEGA328P_32PIN	
1	U2	DS1307	
Diodes			
1	D1	LED-YELLOW	
Miscellaneous			
1	BAT1	3V	
1	LCD1	LM044L	
1	RV1	10k	
1	X1	32768Hz	
4	key		

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

ЗАДАНИЕ

на разработку

Выдано студентам: Авакумову Д.Д., Логинову Д.А. группы ОМРБ-1

Название проекта: Контроллер для управления силовой нагрузкой по расписанию

Назначение: Включение/выключение силовой нагрузки по расписанию.

Область использования: Бытовое применение

Функциональное описание устройства:

Пользователь включает установку. Вводит расписание включения/ выключения силовой нагрузки. Установка включает/выключает силовую нагрузку по расписанию.

Техническое описание устройства: Блок управления, содержащий: устройство отсчёта интервалов времени на основе однокристального микроконтроллера; блок коммутации силовой нагрузки; светодиодный цифровой индикатор для отображения текущего времени; блок питания с выходным напряжением +5В.

Требования: длительность интервала расписания – 1 сутки; количество каналов управления силовой нагрузкой – 1 шт.

План работ:


Наименование работ	Срок
Сбор и изучение материалов, необходимых для проектирования	09.2022
Разработка блок-схемы устройства	09.2022
Выбор элементов, разработка принципиальной схемы устройства	10.2022
Разработка программного кода и моделирование системы	10.2022
Изготовление прототипа устройства	11.2022
Тестирование и финальная отладка	12.2022
Оформление отчета	12.2022

Комментарии:

Перечень графического материала:

1. Блок-схема устройства _____
2. Принципиальная схема устройства _____
3. Блок схемы алгоритмов _____
4. Внешний вид устройства _____

Руководитель проекта

 12.09.2022 В.А. Егоров
(подпись, дата)

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

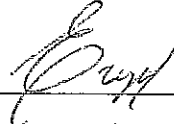


ПАСПОРТ

Аппаратно-программный комплекс

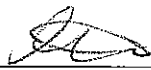
«Контроллер для управления силовой нагрузкой по расписанию»

Руководитель проекта


28.12.2022
(подпись, дата)

В.А. Егоров

Ответственный исполнитель


28.12.2022
(подпись, дата)

Д.А. Логинов

Комсомольск-на-Амуре 2022

Содержание

1	Общие положения.....	7
1.1	Наименование изделия.....	7
1.2	Наименования документов, на основании которых ведется проектирование системы	7
1.3	Перечень организаций, участвующих в разработке системы.....	7
1.4	Сведения об использованных при проектировании нормативно-технических документах.....	8
2	Назначение и принцип действия.....	9
2.1	Назначение изделия	9
2.2	Области использования изделия.....	9
2.3	Принцип действия	9
3	Состав изделия и комплектность	10
4	Технические характеристики	11
4.1	Основные технические характеристики блока.....	11
5	Устройство и описание работы изделия	12
5.1	Устройство изделия.....	12
5.2	Описание работы изделия	13
6	Условия эксплуатации	14
6.1	Правила и особенности размещения изделия	14
6.2	Меры безопасности	15
6.3	Правила хранения и транспортирования	15

1 Общие положения

Настоящий паспорт является документом, предназначенным для ознакомления с основными техническими характеристиками, устройством, правилами установки и эксплуатации устройства «Контроллер для управления силовой нагрузкой по расписанию» (далее «изделие»).

Паспорт входит в комплект поставки изделия. Прежде, чем пользоваться изделием, внимательно изучите правила обращения и порядок работы с ним. В связи с постоянной работой по усовершенствованию изделия, повышающей его надежность и улучшающей условия эксплуатации, в конструкцию могут быть внесены изменения, не отраженные в данном издании.

1.1 Наименование изделия

Полное наименование системы – аппаратно-программный комплекс *«Контроллер для управления силовой нагрузкой по расписанию»* (АПК КУСНР).

1.2 Наименования документов, на основании которых ведется проектирование системы

Создание АПК КУСНР осуществляется на основании требований и положений следующих документов:

- задание на разработку.

1.3 Перечень организаций, участвующих в разработке системы

Заказчиком создания АПК КУСНР является Федеральное государственное бюджетное образовательное учреждение высшего образования «Комсомольский-на-Амуре государственный университет» (далее заказчик), находящийся по адресу: 681013, Хабаровский край, г. Комсомольск-на-Амуре, Ленина пр-кт., д. 17.

Исполнителем работы по созданию АПК КУСНР являются студенты группы 0МРБ-1 Д.Д. Авакумов, Д.А. Логинов

1.4 Сведения об использованных при проектировании нормативно-технических документах

При проектировании использованы следующие нормативно-технические документы:

ГОСТ 2.001-2013. Единая система конструкторской документации. Общие положения.

ГОСТ 2.102-2013. Единая система конструкторской документации. Виды и комплектность конструкторских документов.

ГОСТ 2.105-95. Единая система конструкторской документации. Общие требования к текстовым документам.

ГОСТ 2.610-2006. Единая система конструкторской документации. Правила выполнения эксплуатационных документов.

ГОСТ 2.004-88. Единая система конструкторской документации. Общие требования к выполнению конструкторских технологических документов на печатающих и графических устройствах вывода ЭВМ.

ГОСТ 2.051-2006. Единая система конструкторской документации. Электронные документы. Общие положения.

ГОСТ 2.052-2006. Единая система конструкторской документации. Электронная модель изделия. Общие положения.

ГОСТ 2.601-2013. Единая система конструкторской документации. Эксплуатационные документы.

2 Назначение и принцип действия

2.1 Назначение изделия

АПК КУСНР – представляет из себя устройство отсчёта интервалов времени на основе однокристалльного микроконтроллера и блок коммутации силовой нагрузки.

В состав изделия входят: Блок управления, содержащий: устройство отсчёта интервалов времени на основе однокристалльного микроконтроллера; блок коммутации силовой нагрузки; светодиодный цифровой индикатор для отображения текущего времени; блок питания с выходным напряжением +5В.

2.2 Области использования изделия

Изделие может использоваться для коммутации силовой нагрузки по расписанию.

2.3 Принцип действия

Пользователь включает установку. Вводит расписание включения/выключения силовой нагрузки. Установка включает/выключает силовую нагрузку по расписанию.

3 Состав изделия и комплектность

В комплект поставки входит:

- *Плата устройства*
- *Сетевой питающий адаптер*
- *Паспорт.*

4 Технические характеристики

4.1 Основные технические характеристики блока

Основные технические характеристики приведены в таблице 1.

Таблица 1 - Технические характеристики блока

Наименование параметра	Значение
Потребляемый ток, мА	100
Диапазон рабочих температур прибора, С°	-55 ... +125
Длительность интервала расписания	24 часа
Количество каналов управления силовой нагрузкой	1
Питание, В	5
Габариты, мм	
Масса нетто, кг	0.2

5 Устройство и описание работы изделия

5.1 Устройство изделия

Структурная схема изделия представлена на рисунке 1.



Рисунок 1 – Структурная схема изделия

В состав изделия входят:

- Кнопки ввода расписания и коррекции времени;
- Блок управления;
- Блок коммутации силовой нагрузки;
- Алфавитно-цифровой индикатор;
- Блок питания;

Кнопки ввода расписания и коррекции времени позволяют установить текущее время и ввести расписание коммутации силовой нагрузки.

Блок управления представляет из себя модуль ARDUINO-NANO, на основе микроконтроллера ATM328P:

– выполняет опрос текущего времени из микросхемы часов DS1307 по интерфейсу I2C;

– сравнивает текущее время с расписанием включения и выключения нагрузки;

– при наличии совпадения, коммутирует силовую нагрузку переменного тока;

Блок коммутации силовой нагрузки выполнен на основе оптосимистора.

Алфавитно-цифровой индикатор – служит для контроля коррекции времени и ввода расписания коммутации нагрузки.

Блок питания преобразует однофазное переменное напряжение 220V в постоянное напряжение +5 V для питания схемы устройства.

5.2 Описание работы изделия

Пользователь включает установку. Вводит расписание включения/выключения силовой нагрузки. Установка включает/выключает силовую нагрузку по расписанию.

Блок-схема работы управляющей программы приведена в Приложении А.

6 Условия эксплуатации

Изделие выпускается в климатическом исполнении УХЛ 4.2 по ГОСТ 15150-69 и предназначен для использования в стационарных условиях в закрытых помещениях при соответствующих климатических условиях:

- интервал температур от +10 до +35 °С;
- относительная влажность воздуха до 80 % при температуре +25 °С;
- высота над уровнем моря не более 2000 м;
- атмосферное давление от 86,6 до 106 кПа (от 650 до 800 мм рт. ст.).

В помещении, где используется изделие не должно возникать условий для конденсации влаги (выпадения росы). Изделие является электронным прибором, требующим бережного обращения.

Для обеспечения безотказной работы, сохранения точности и его сбережения необходимо соблюдать следующие правила:

- изучить паспорт, прежде чем приступить к работе с изделием;
- предохранять изделие от ударов и повреждений;
- не допускать самостоятельную разборку изделия.

6.1 Правила и особенности размещения изделия

ВНИМАНИЕ! При эксплуатации изделия запрещается проводить самостоятельно какие-то либо работы по извлечению и установке внутренних компонентов изделия.

6.2 Меры безопасности

Необходимо соблюдать требования техники безопасности и следующие меры предосторожности:

- не оставлять изделие включенным без наблюдения;
- после транспортировки в холодное время года изделие необходимо выдержать при комнатной температуре не менее двух часов;
- внутренние осмотры и ремонт изделия должны производиться только квалифицированными специалистами;
- не устанавливайте изделие на неустойчивой подставке, стойке или ненадежном кронштейне.

6.3 Правила хранения и транспортирования

Транспортирование изделия в упакованном виде может производиться железнодорожным, автомобильным (в закрытых транспортных средствах), воздушным, речным и морским видами транспорта в соответствии с правилами перевозок грузов, действующих на транспорт данного вида. Условия транспортирования изделия по части воздействия климатических факторов должны соответствовать группе 5 по ГОСТ 15150.

После транспортирования изделие должно быть выдержано не менее 2 часов в транспортной таре при температуре 20 ± 5 °С и относительной влажности воздуха не более 80 %.

Распакованное изделие должно храниться в отапливаемом и вентилируемом чистом помещении при температуре от +5 до +40 °С и относительной влажности воздуха не более 60 %. При температуре ниже 25 °С допускается увеличение относительной влажности до 80 %. Воздух в помещении не должен содержать примесей, вызывающих коррозию металлов, налеты на поверхностях оптических деталей.

ПРИЛОЖЕНИЕ А

(обязательное)

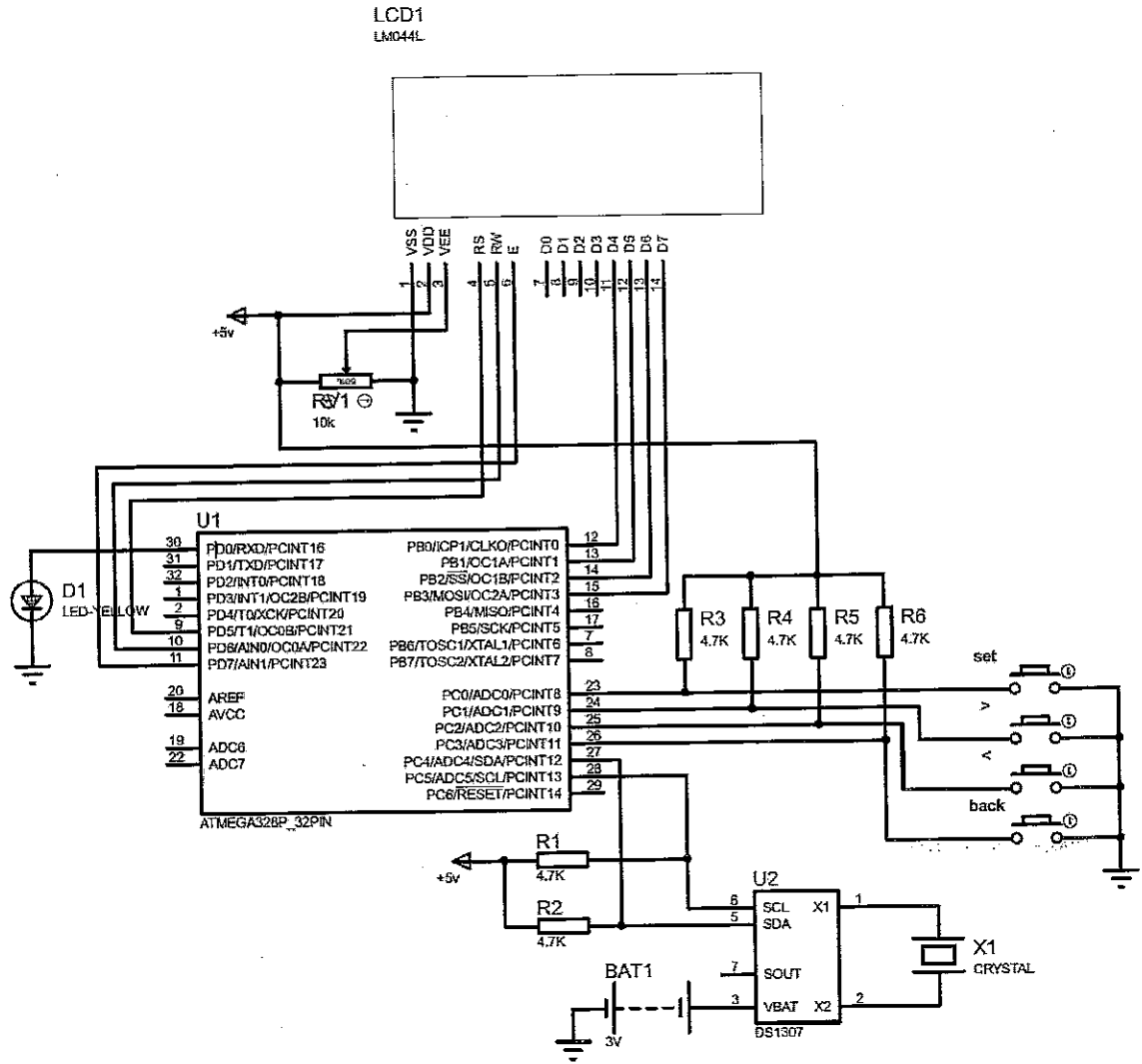


Рисунок А1 - Принципиальная схема изделия

Спецификация к принципиальной схеме			
Поз.обознач.	Наименование	Кол.	Примечание
	Светодиоды		
D7	LED_US	1	
	Резисторы		
R1-R6	4.7 КОм	6	
	Интегральные схемы		
U2	DS 1307	1	
	Модули		
M1	ARDUINO_NANO	1	
	Разное		
BAT1	3V	1	
LCD1	LM044L	1	
X1	32768Hz	1	
Key		4	
RV1	10K	1	Digikey 3252W-203LF-ND

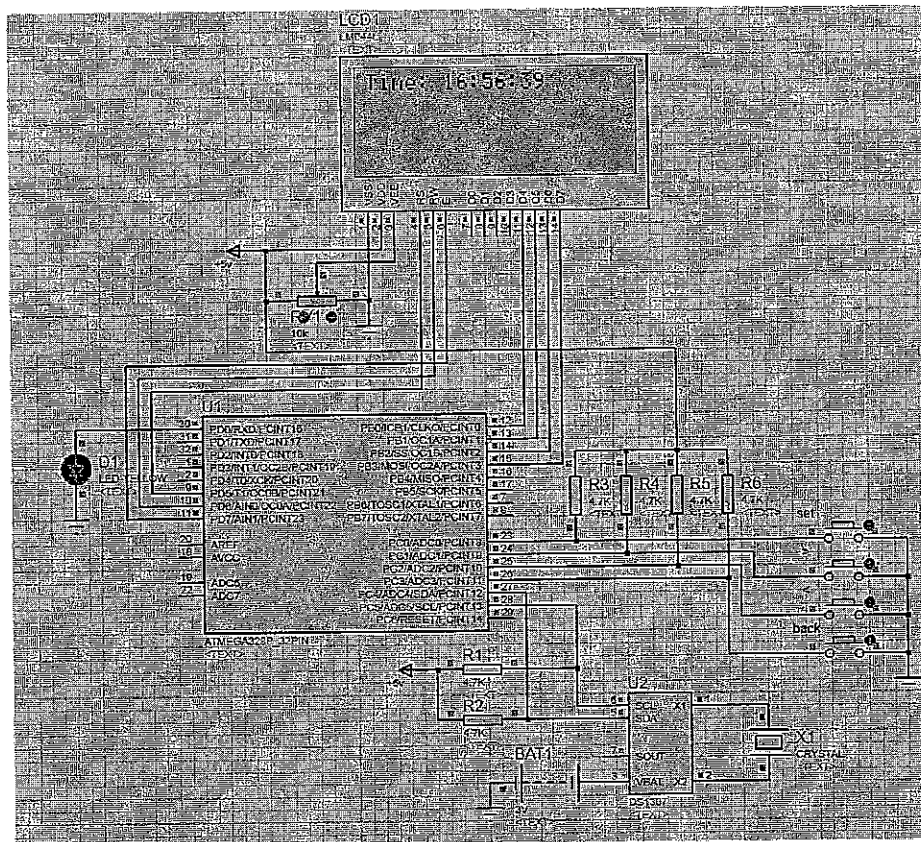


Рисунок А2 - Модель изделия в Proteus (отображение текущего времени)

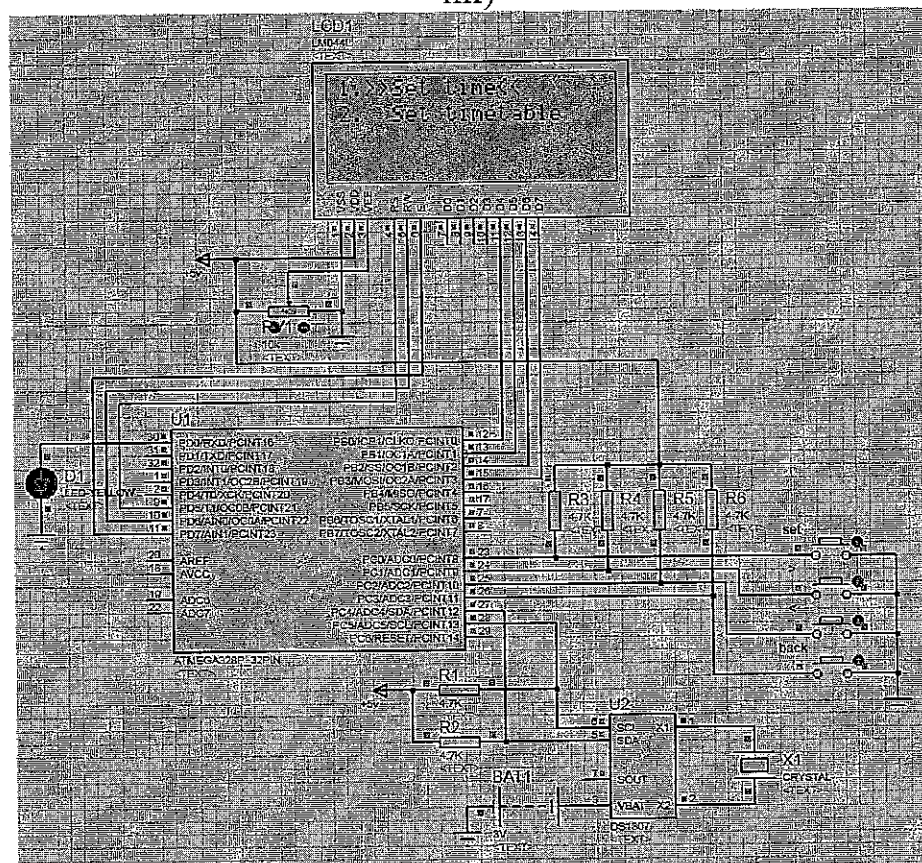


Рисунок А3 - Модель изделия в Proteus (выбор главного меню)

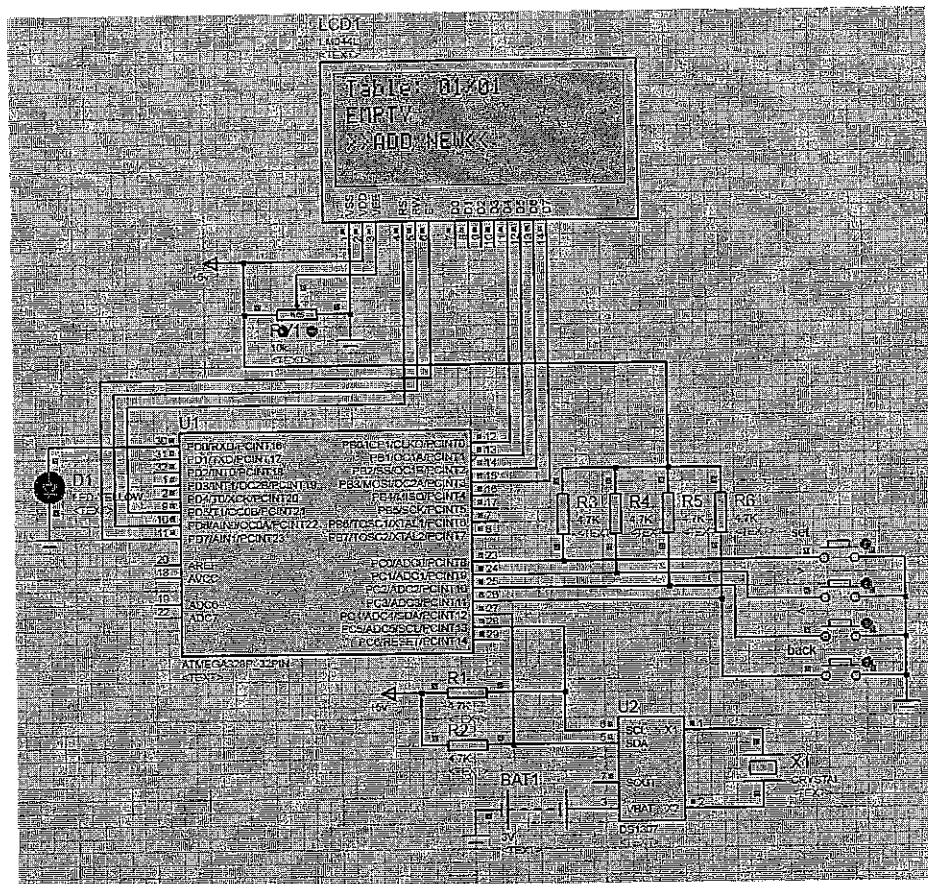


Рисунок А4 - Модель изделия в Proteus (меню создания нового расписания)

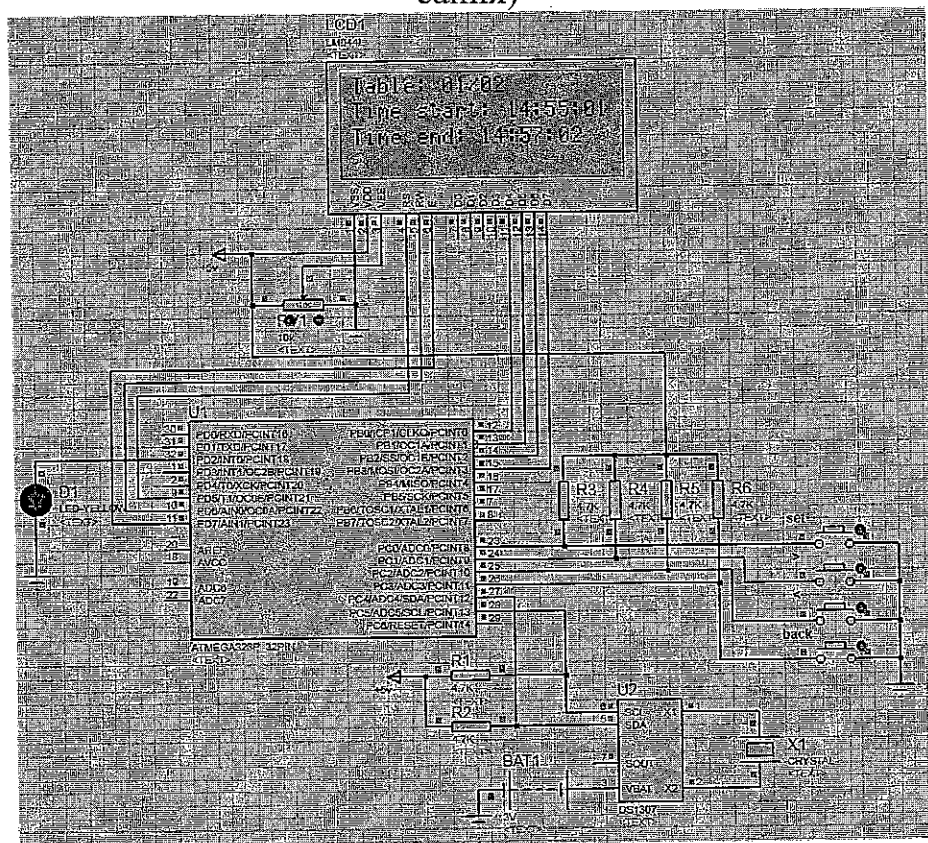


Рисунок А5 – Модель изделия в Proteus (исходное состояние, время включения и время выключения нагрузки установлено)



Рисунок А6 – Модель изделия в Proteus (время включения нагрузки превышено, нагрузка включена)

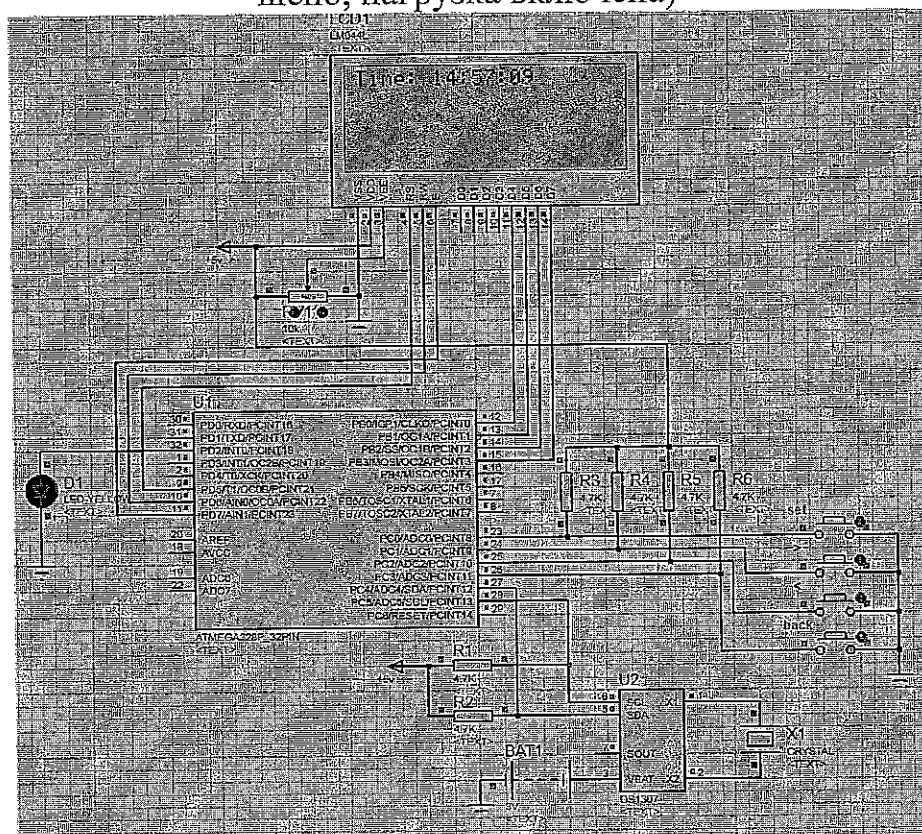


Рисунок А7 – Модель изделия в Proteus (время выключения нагрузки превышено, нагрузка выключена)

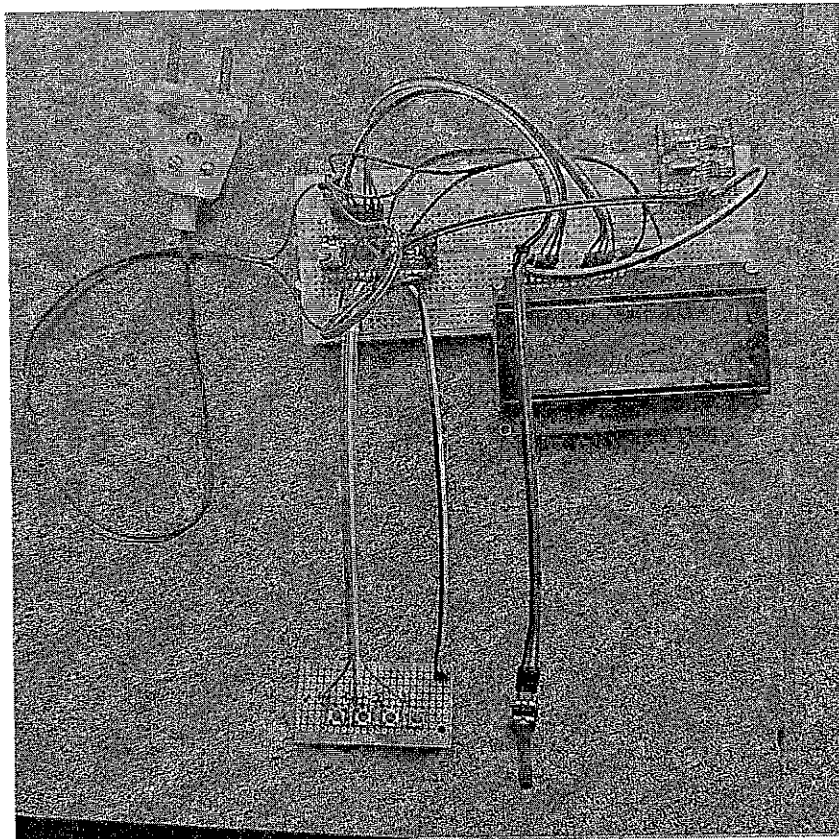


Рисунок А9 – Внешний вид макета изделия

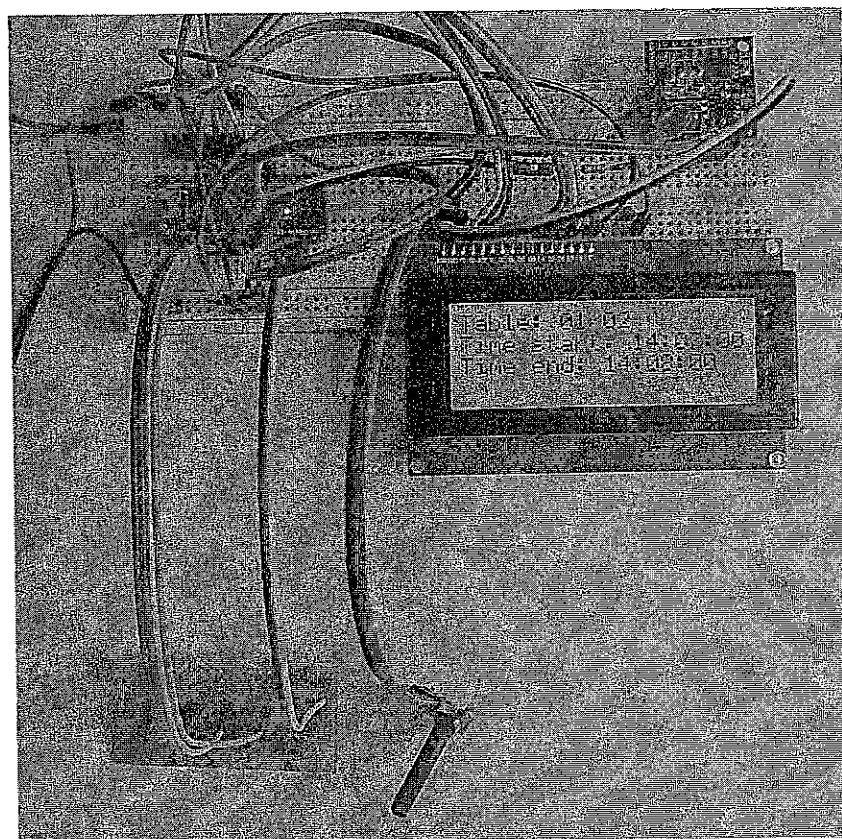


Рисунок А10 – Тестирование изделия (исходное состояние: -текущее время 14:06:42 ; -время включения нагрузки 14:07:00; -время выключения нагрузки 14:08:00)

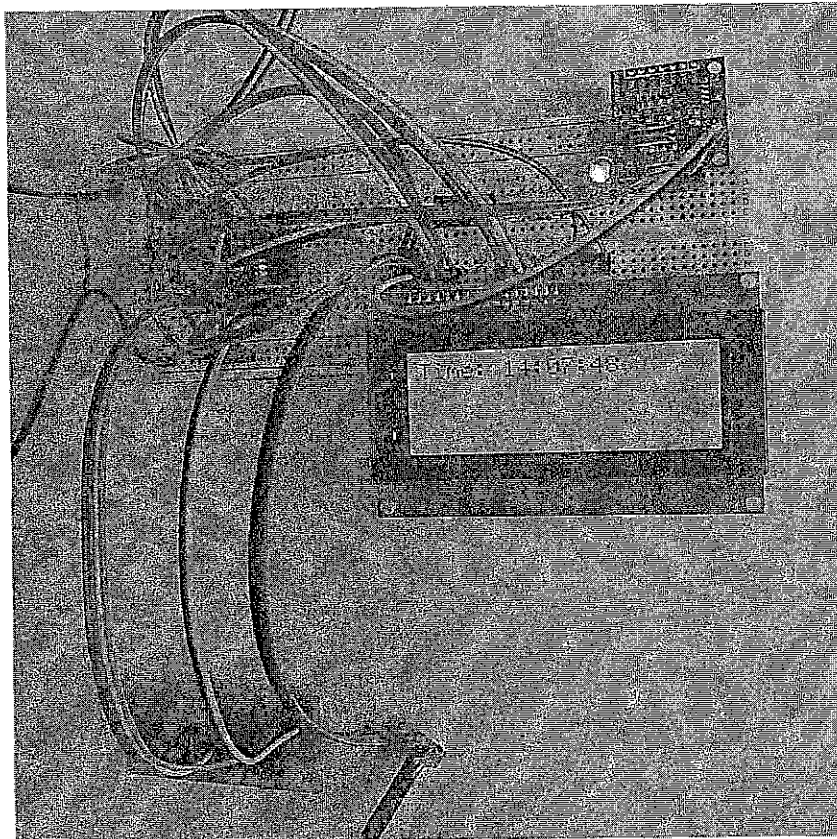


Рисунок А11 – Тестирование изделия (-текущее время 14:07:48; -время включения нагрузки 14:07:00, нагрузка включена(красный светодиод горит))

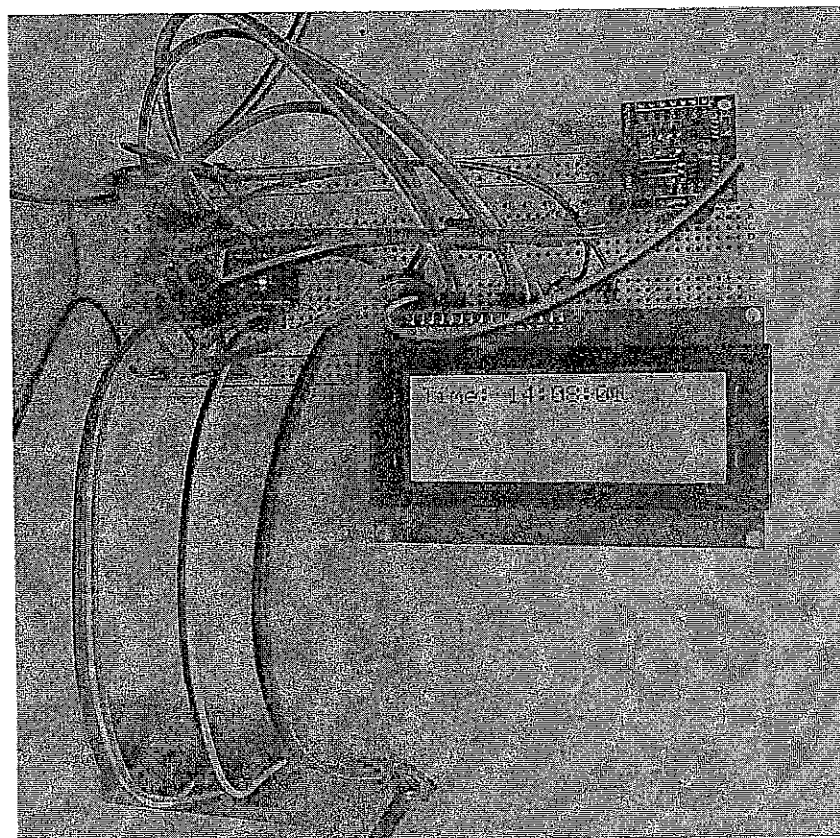


Рисунок А12 – Тестирование изделия (-текущее время 14:08:01; -время выключения нагрузки: 14:08:00, нагрузка выключена (красный светодиод не горит))

Порядок настройки режимов работы изделия

Кнопки:

1. SET – кнопка выбора текущего действия / кнопка установки времени;



2. < - кнопка уменьшения числа / кнопка «шага» по списку вверх;



3. > - кнопка увеличения числа / кнопка «шага» по списку вниз;

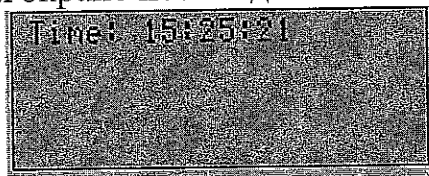


4. Back – кнопка возврата из любого состояния экрана на предыдущий экран.



1. Установки времени.

Находясь на стартовом экране необходимо нажать кнопку SET.



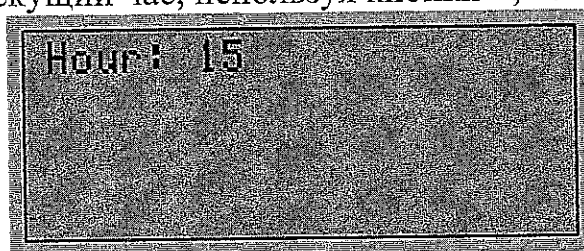
Далее с помощью кнопок <, >.

Выбираем в меню установку времени (Set time), и нажимаем кнопку

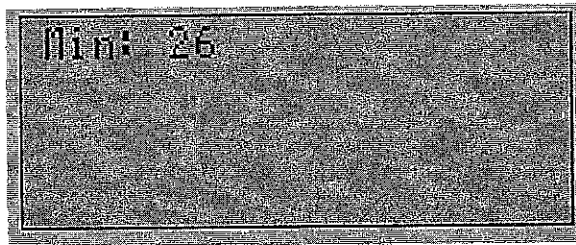
SET



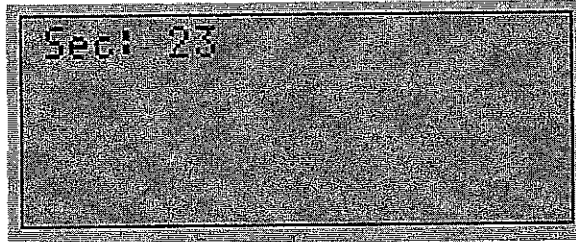
Устанавливаем текущий час, используя кнопки <, >.



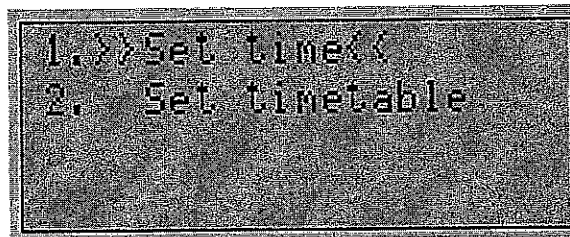
После установки нажимаем кнопку SET, и переходим в установку текущих минут



Устанавливаем текущую минуту, используя кнопки <, >.
После установки нажимаем кнопку SET, и переходим в установку текущих секунд



После этого нажимаем кнопку SET, устанавливаем секунды и выходим в меню



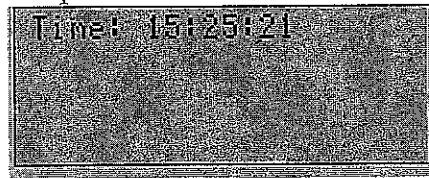
Далее нажимаем кнопку BACK и возвращаемся на стартовый экран с новым, установленным временем



2. Установка расписания.

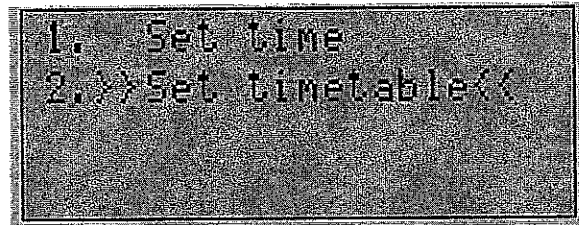
2.1 Установка нового расписания в пустой лист расписания.

Находясь на стартовом экране необходимо нажать кнопку SET.



Далее с помощью кнопок <, >.

Выбираем в меню установку времени (Set timetable), и нажимаем кнопку SET

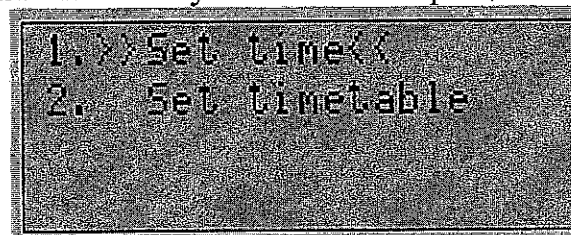


В открывшемся окне,

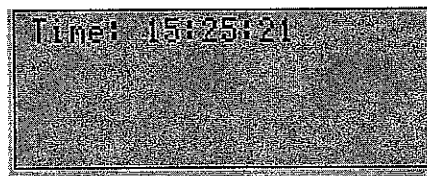


нажимаем ещё раз кнопку SET и переходим на установку времени для нового расписания (п. 2.2).

После этого нажимаем кнопку BACK и возвращаемся в меню

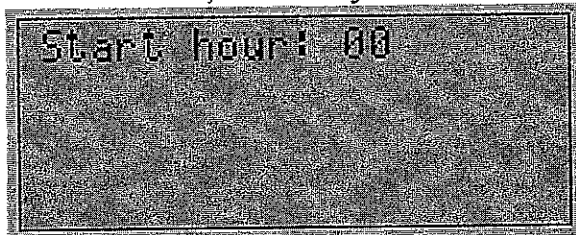


Далее снова нажимаем кнопку BACK и возвращаемся на стартовый экран

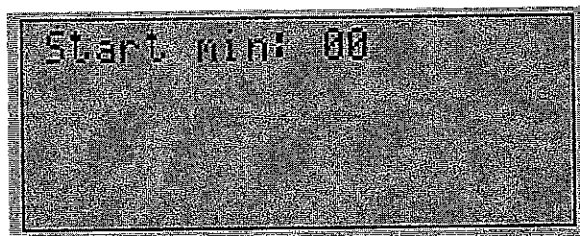


2.2 Установка времени для нового расписания

Устанавливаем начальный час, используя кнопки <, >.

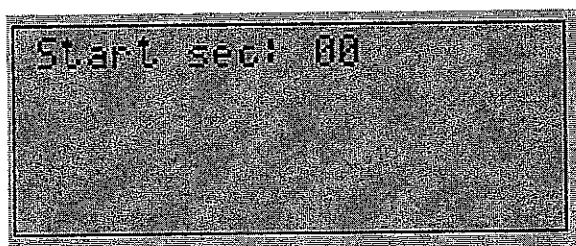


После установки нажимаем кнопку SET, и переходим в установку начальных минут



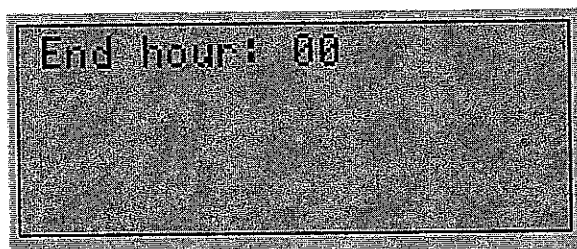
Устанавливаем начальную минуту, используя кнопки <, >.

После установки нажимаем кнопку SET, и переходим в установку начальных секунд



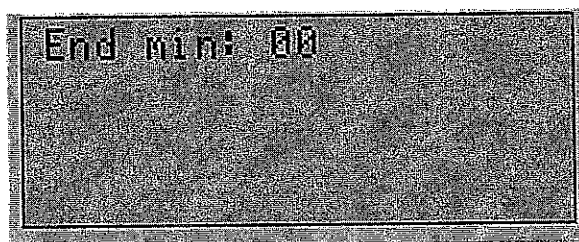
Устанавливаем начальную секунду, используя кнопки <, >.

После этого нажимаем кнопку SET, и переходим в установку конечного часа



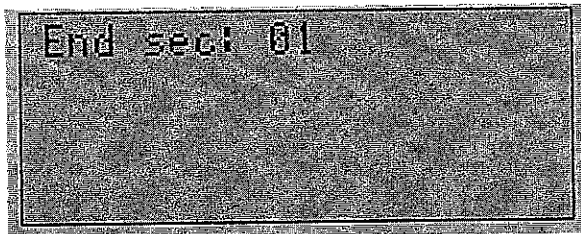
Устанавливаем конечный час, используя кнопки <, >.

После установки нажимаем кнопку SET, и переходим в установку конечных минут



Устанавливаем конечную минуту, используя кнопки <, >.

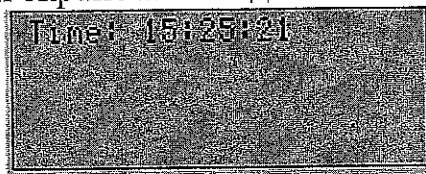
После установки нажимаем кнопку SET, и переходим в установку конечных секунд



Устанавливаем конечную секунду, используя кнопки <, >.
После установки нажимаем кнопку SET, и переходим в таблицу расписания.

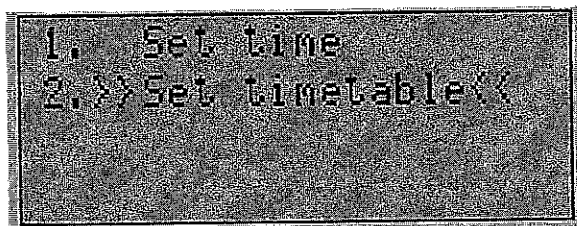
2.3 Установка нового расписания в не пустой лист расписания.

Находясь на стартовом экране необходимо нажать кнопку SET.

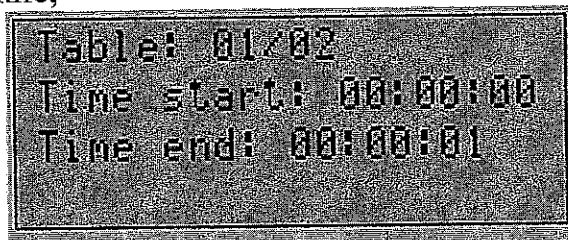


Далее с помощью кнопок <, >.

Выбираем в меню установку времени (Set timetable), и нажимаем кнопку SET



В открывшемся окне,



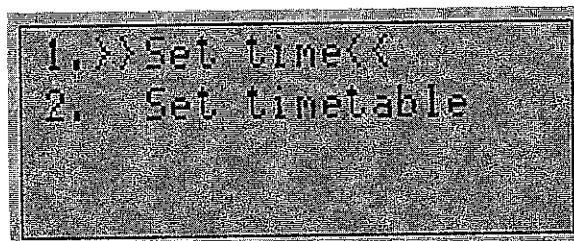
с помощью кнопок <, >.

Выбираем в таблице расписаний последнее на данный момент пустое расписание



Далее нажимаем кнопку SET и переходим на установку времени для нового расписания (п. 2.2).

После этого нажимаем кнопку BACK и возвращаемся в меню



Далее снова нажимаем кнопку BACK и возвращаемся на стартовый экран



2.4 Коррекция времени для уже установленного расписания

Находясь на стартовом экране необходимо нажать кнопку SET.

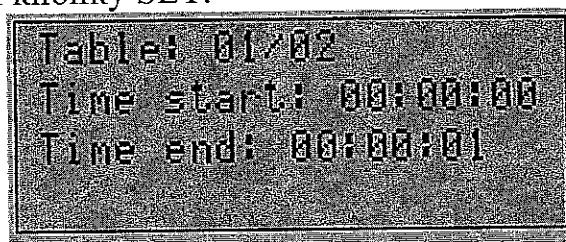


Далее с помощью кнопок <, >.

Выбираем в меню установку времени (Set timetable), и нажимаем кнопку SET.

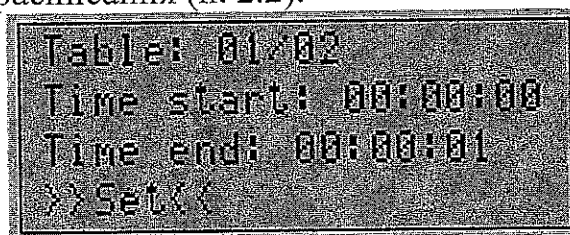


В открывшемся окне, с помощью кнопок <, > выбираем необходимое расписание и нажимаем кнопку SET.



Далее с помощью кнопок <, >.

Выбираем пункт Set, и нажимаем кнопку SET и переходим на установку времени для нового расписания (п. 2.2).



Далее после установки времени, коррекция времени для расписания завершена.

2.4 Коррекция времени для уже установленного расписания

Находясь на стартовом экране необходимо нажать кнопку SET.

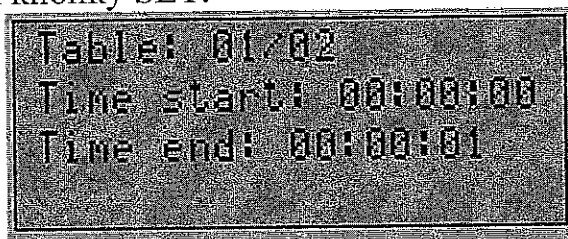


Далее с помощью кнопок <, >.

Выбираем в меню установку времени (Set timetable), и нажимаем кнопку SET.



В открывшемся окне, с помощью кнопок <, > выбираем необходимое расписание и нажимаем кнопку SET.



Далее с помощью кнопок <, >.

Выбираем пункт Delete, и нажимаем кнопку SET.



После нажатия кнопки SET на пункте Delete выбранное расписание удаляется, а вся таблица расписания сдвигается на один пункт вниз.

Описание алгоритмов в блок-схемах



Рисунок А5 - Блок схема void main

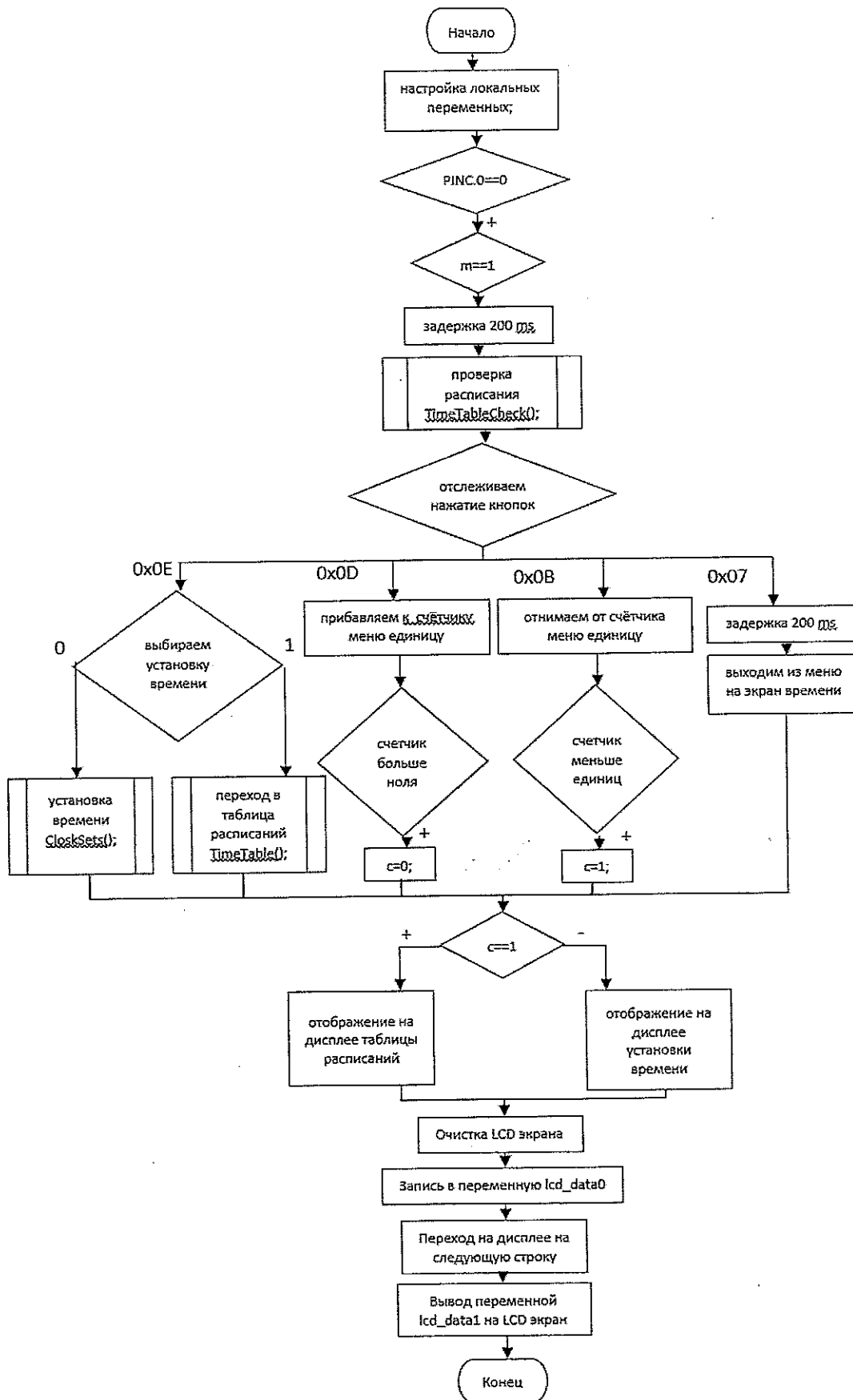


Рисунок А6 – Блок схема void menu

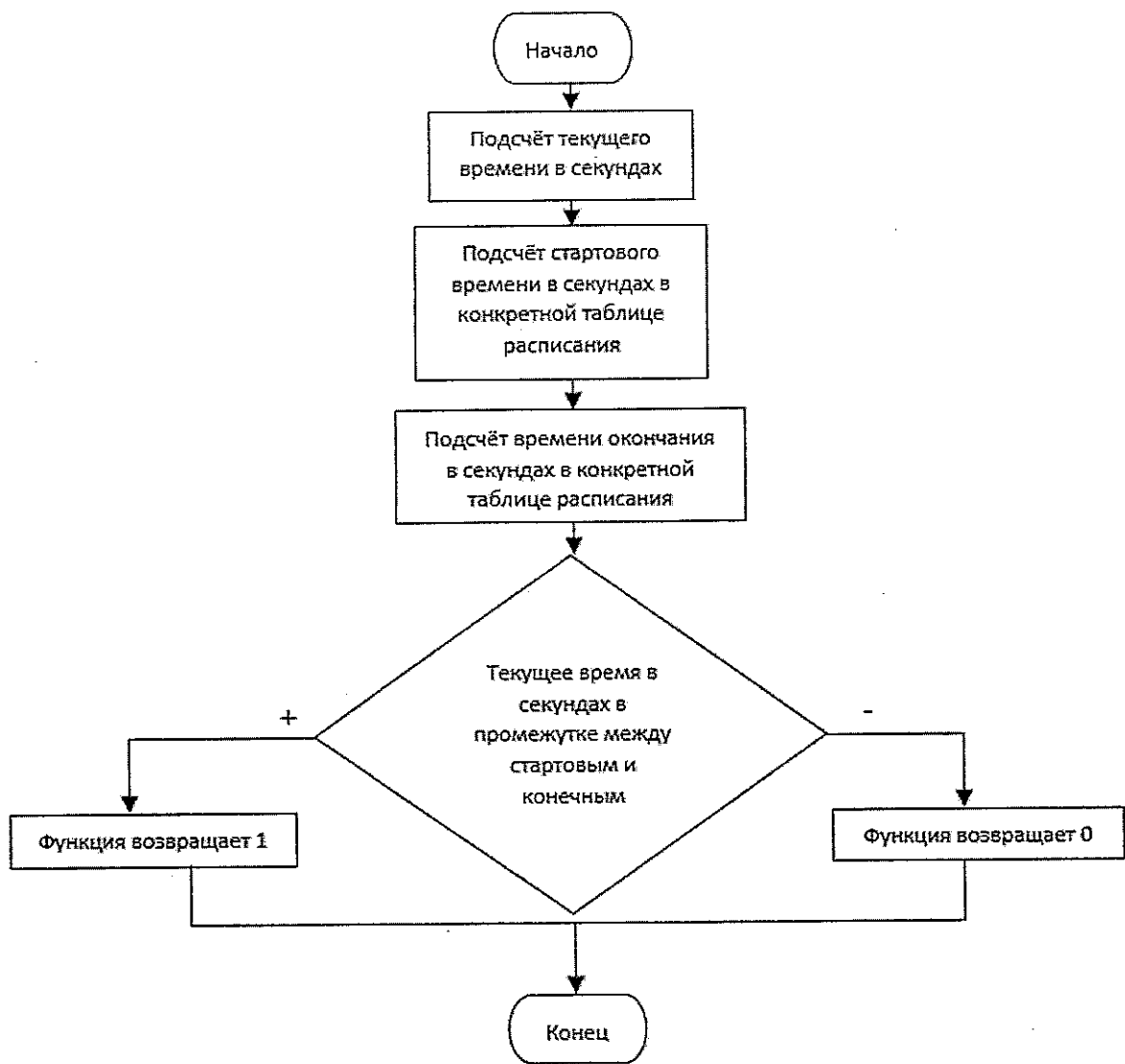


Рисунок А7 - Блок схема `unsigned char TimeCheck(unsigned char current)`

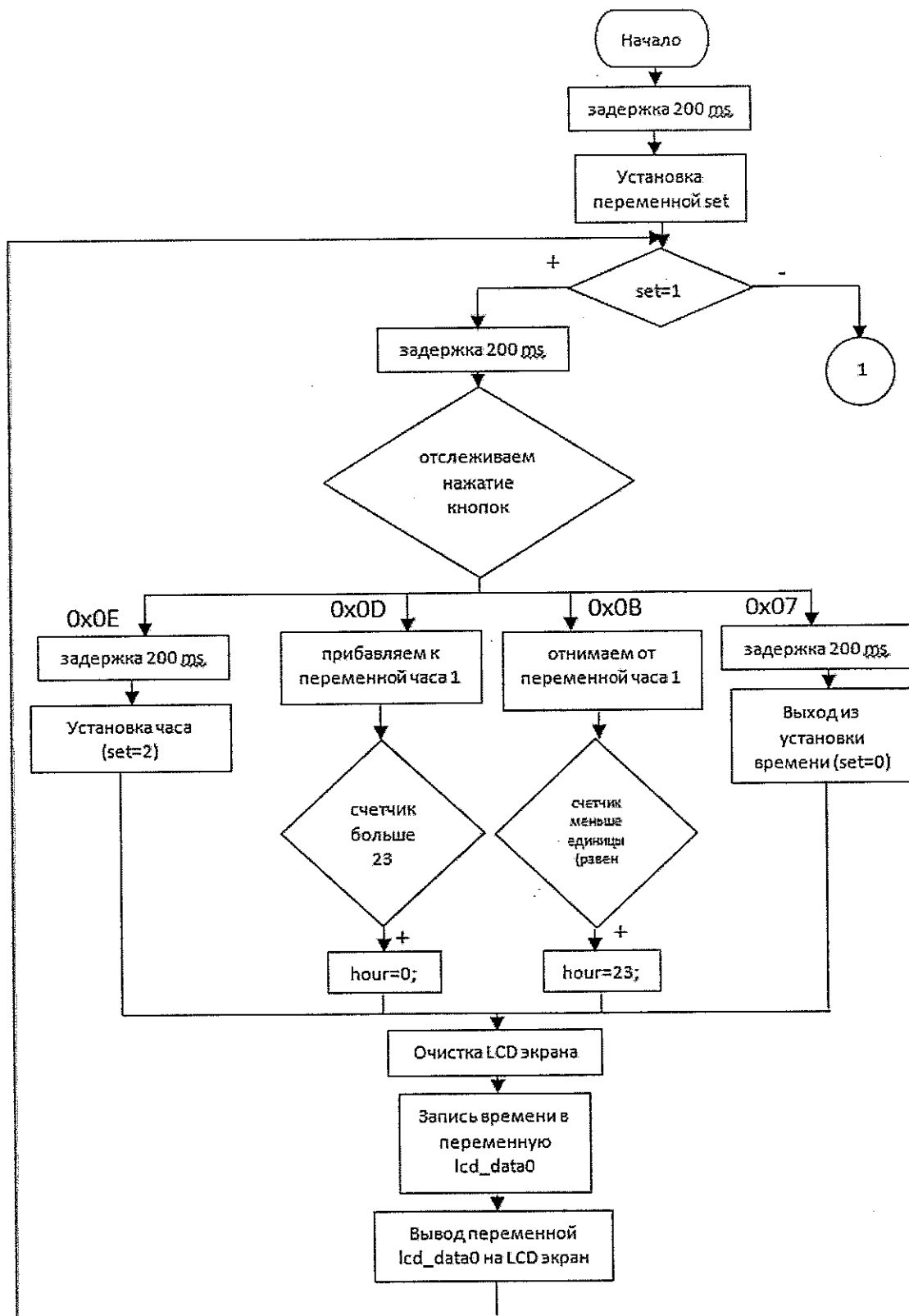


Рисунок А8 - Блок схема void ClookSets(void)

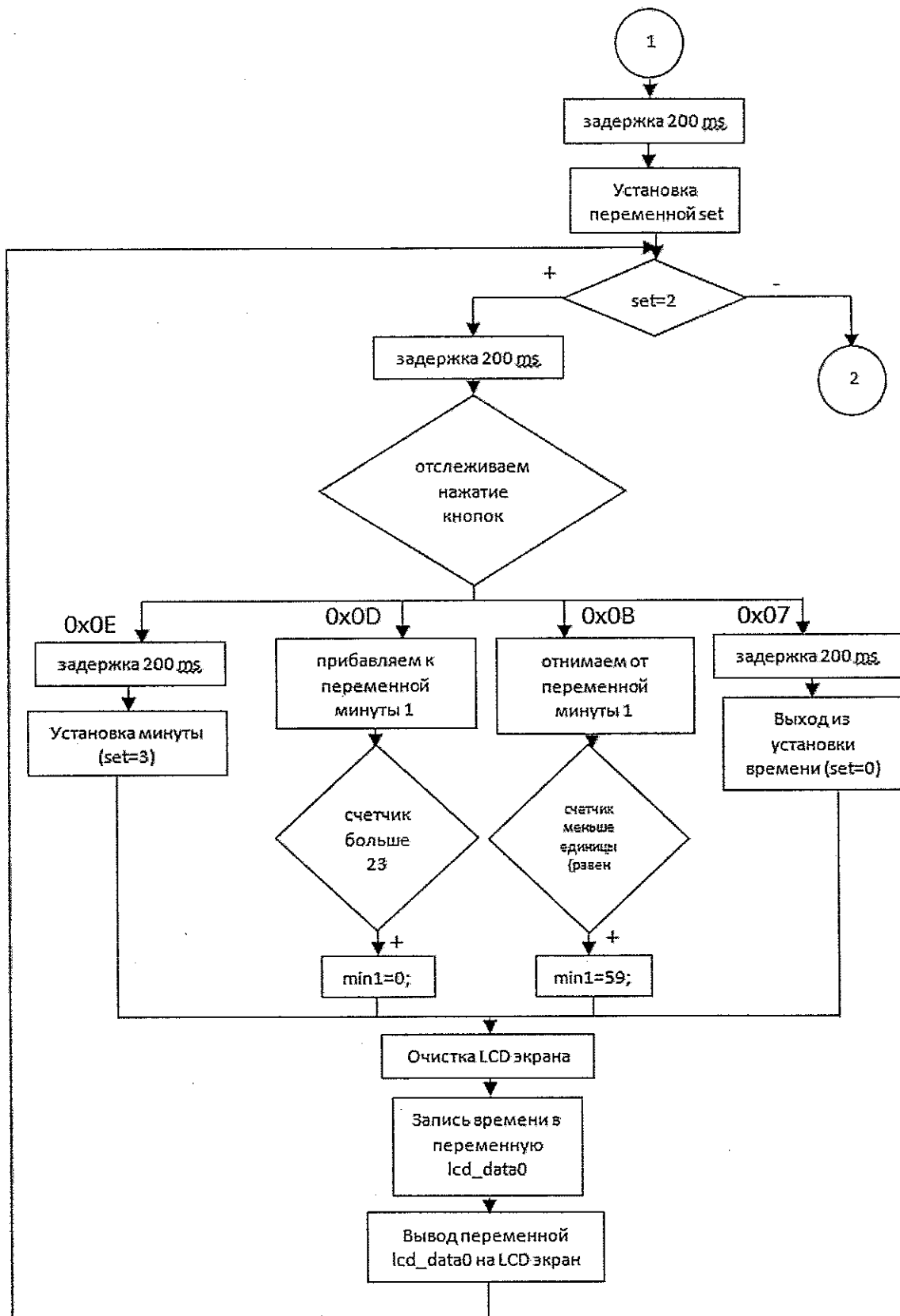


Рисунок А9 - Продолжение блок схемы А8

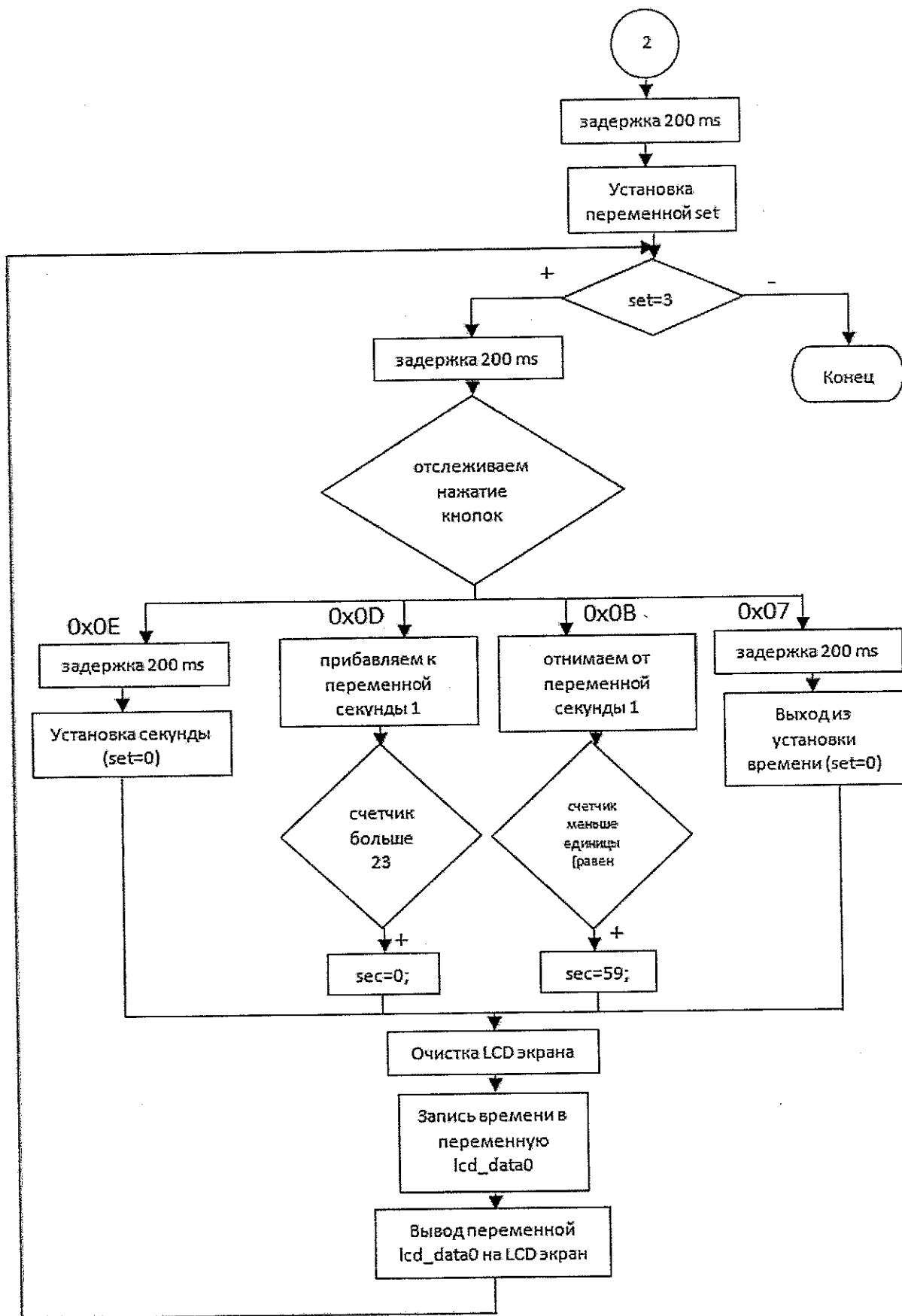


Рисунок А10 - Продолжение блок схемы А9

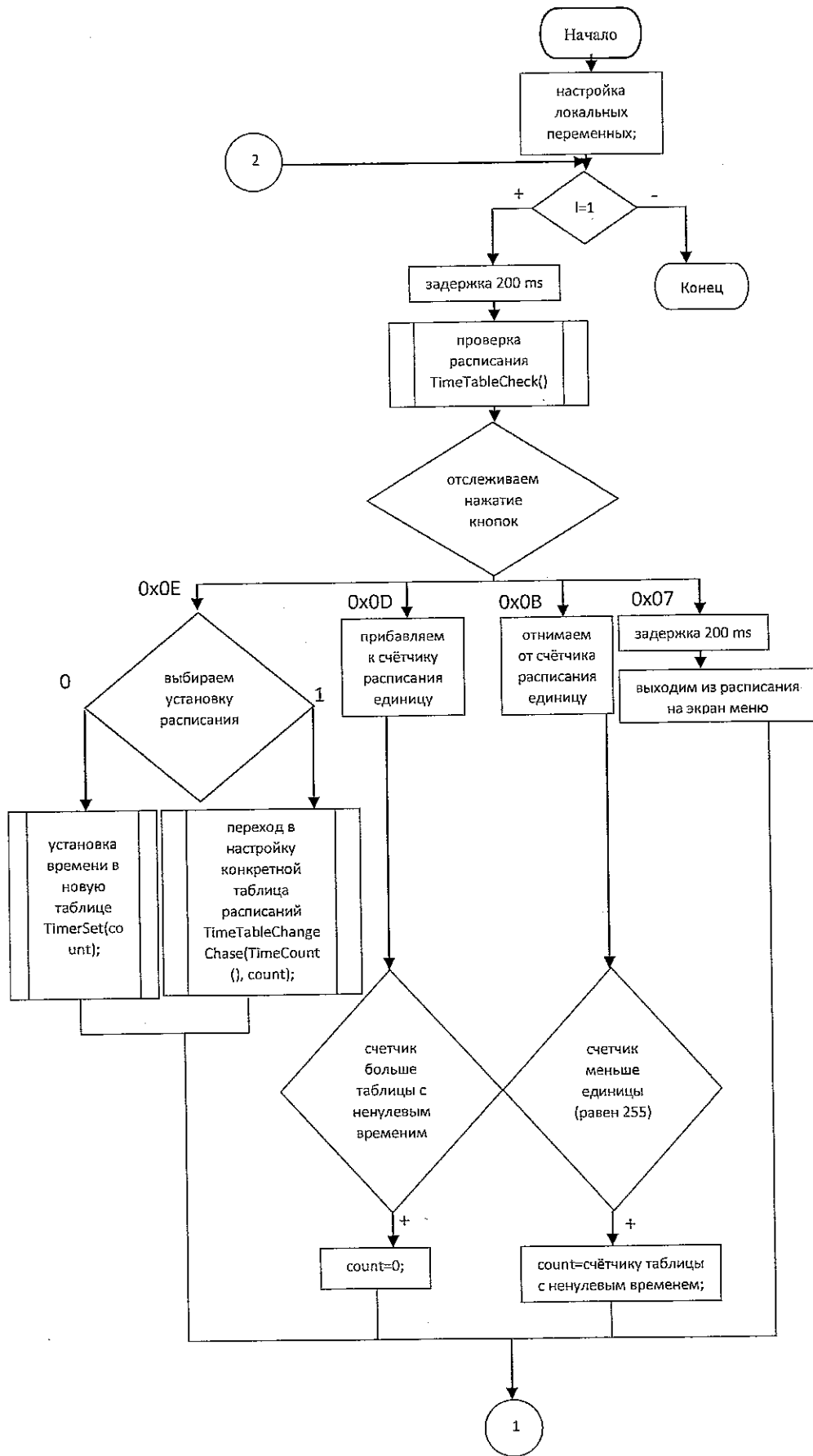


Рисунок А11 - Блок схема void TimeTable()

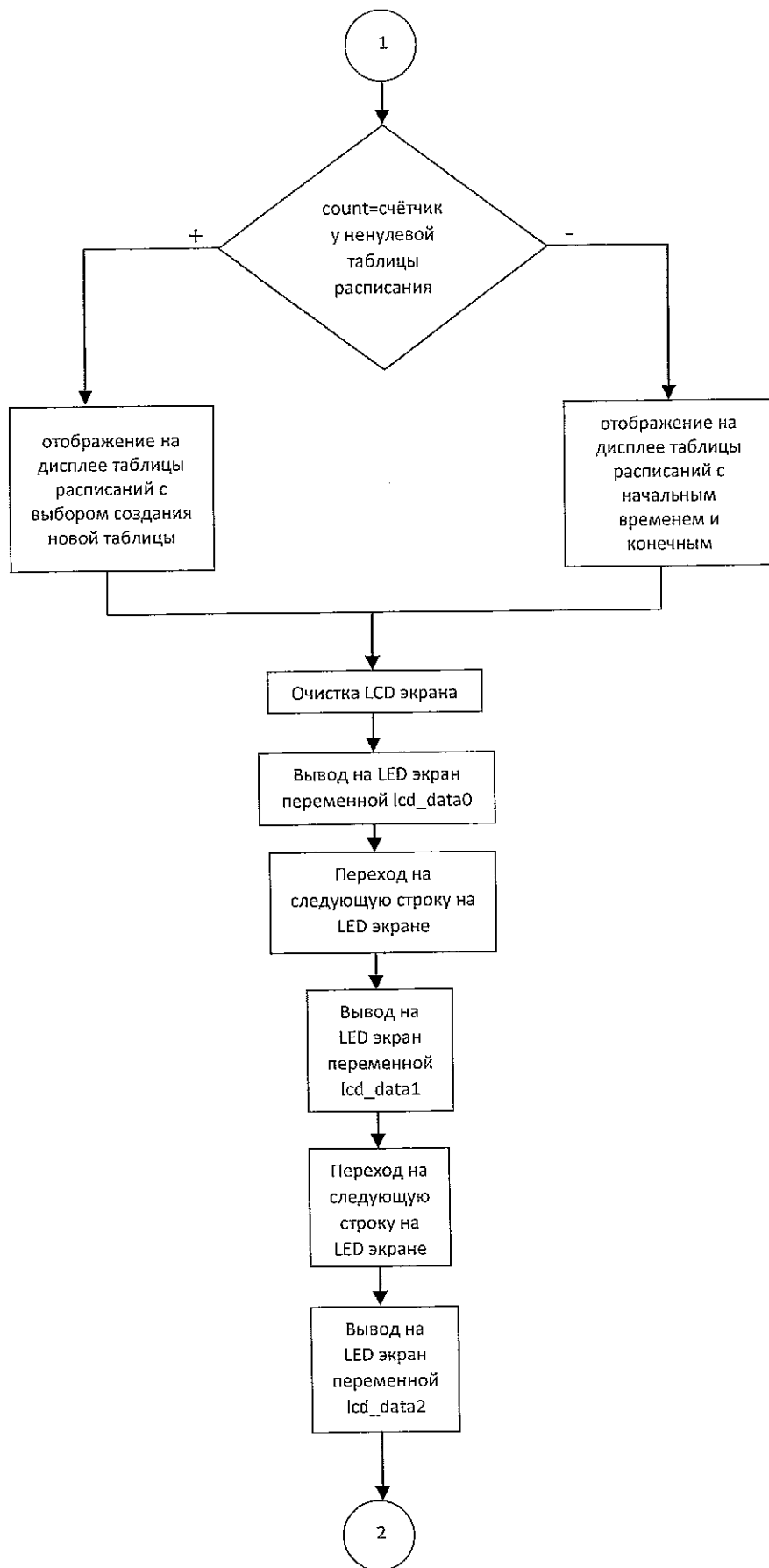


Рисунок А12 - продолжение блок схемы А11

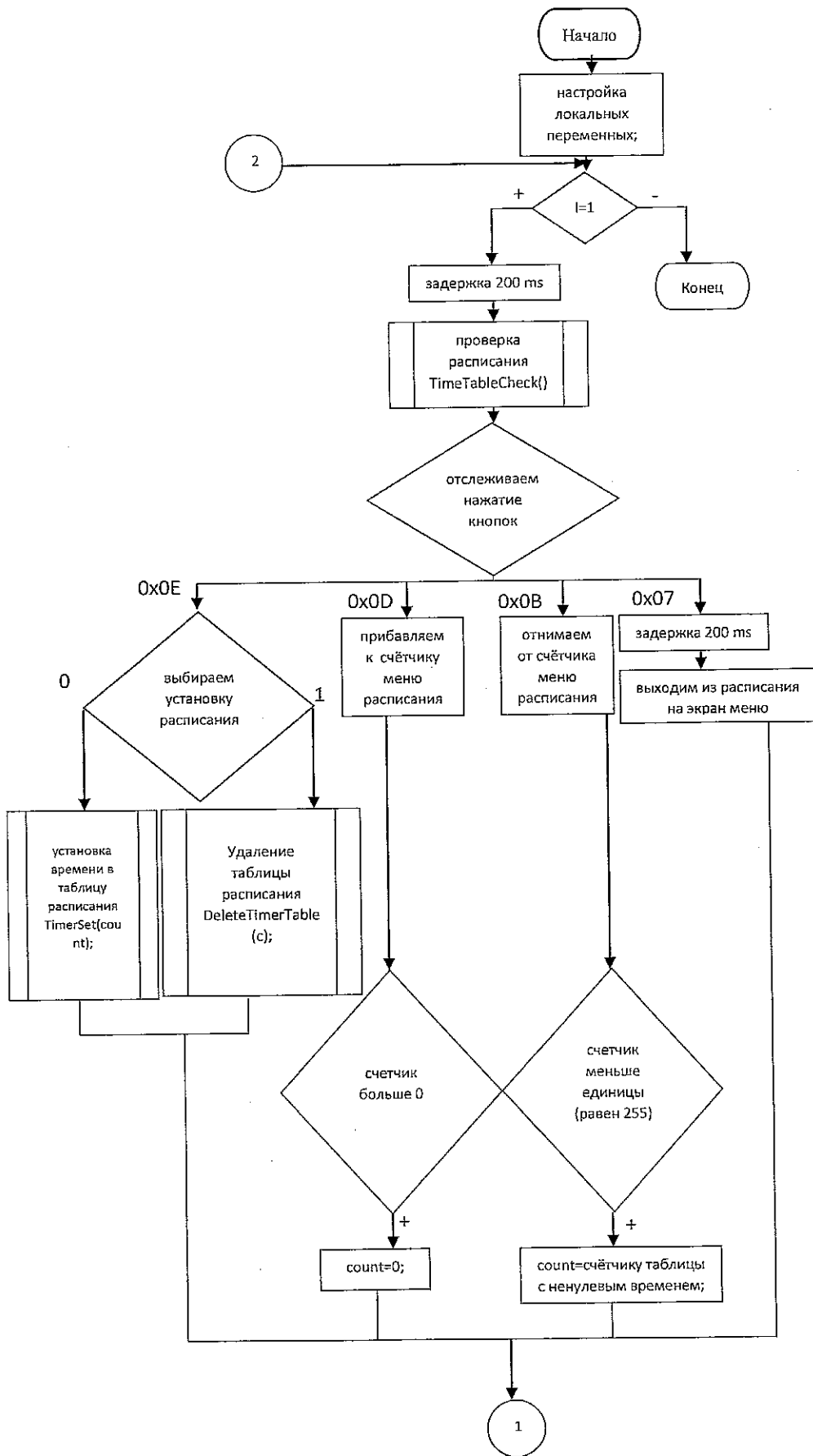


Рисунок А13 - void TimeTableChangeChase (int v, unsigned char c)

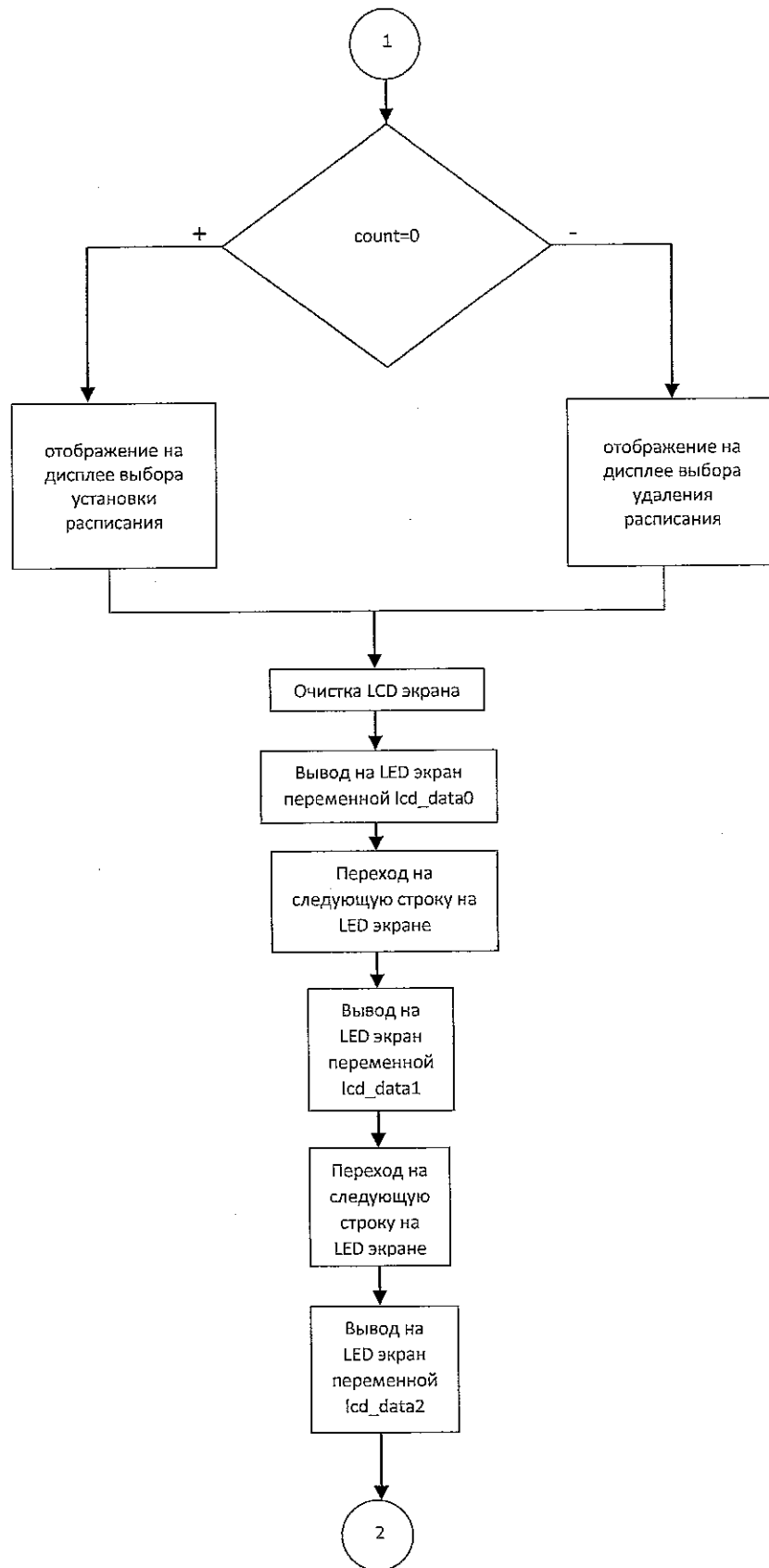


Рисунок А14 - Продолжение блок схемы А13

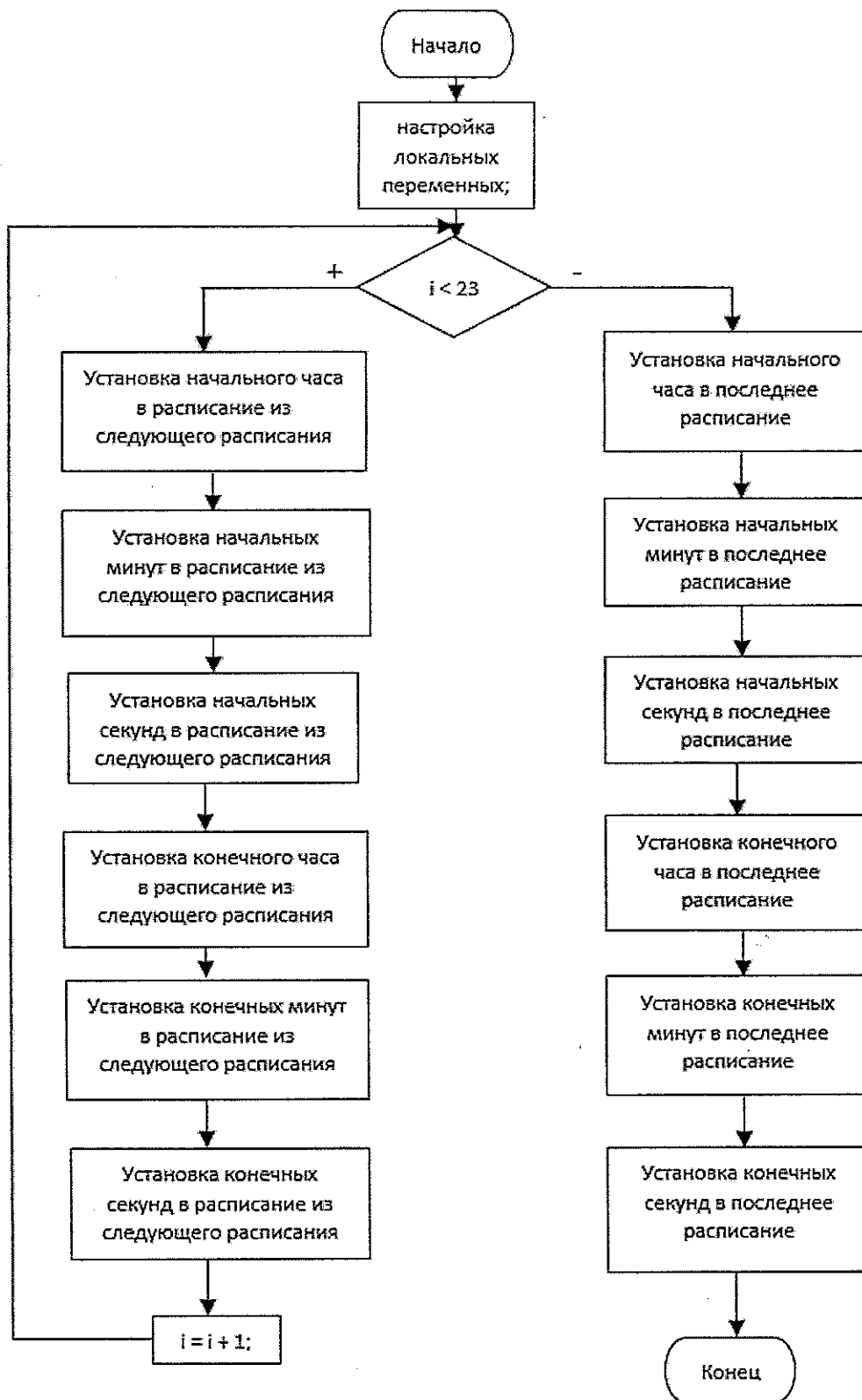


Рисунок А15 - Блок схема void DeleteTimerTable (unsigned char count)

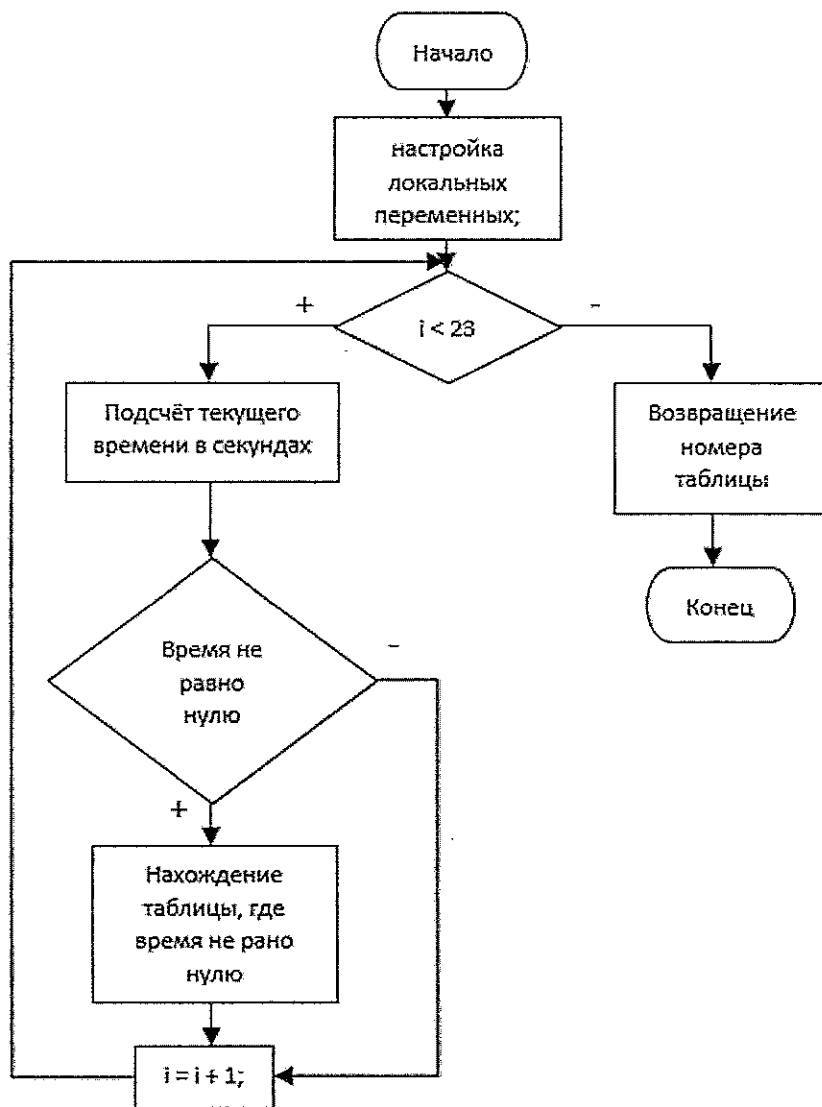


Рисунок А16 - Блок схема int TimeCount()

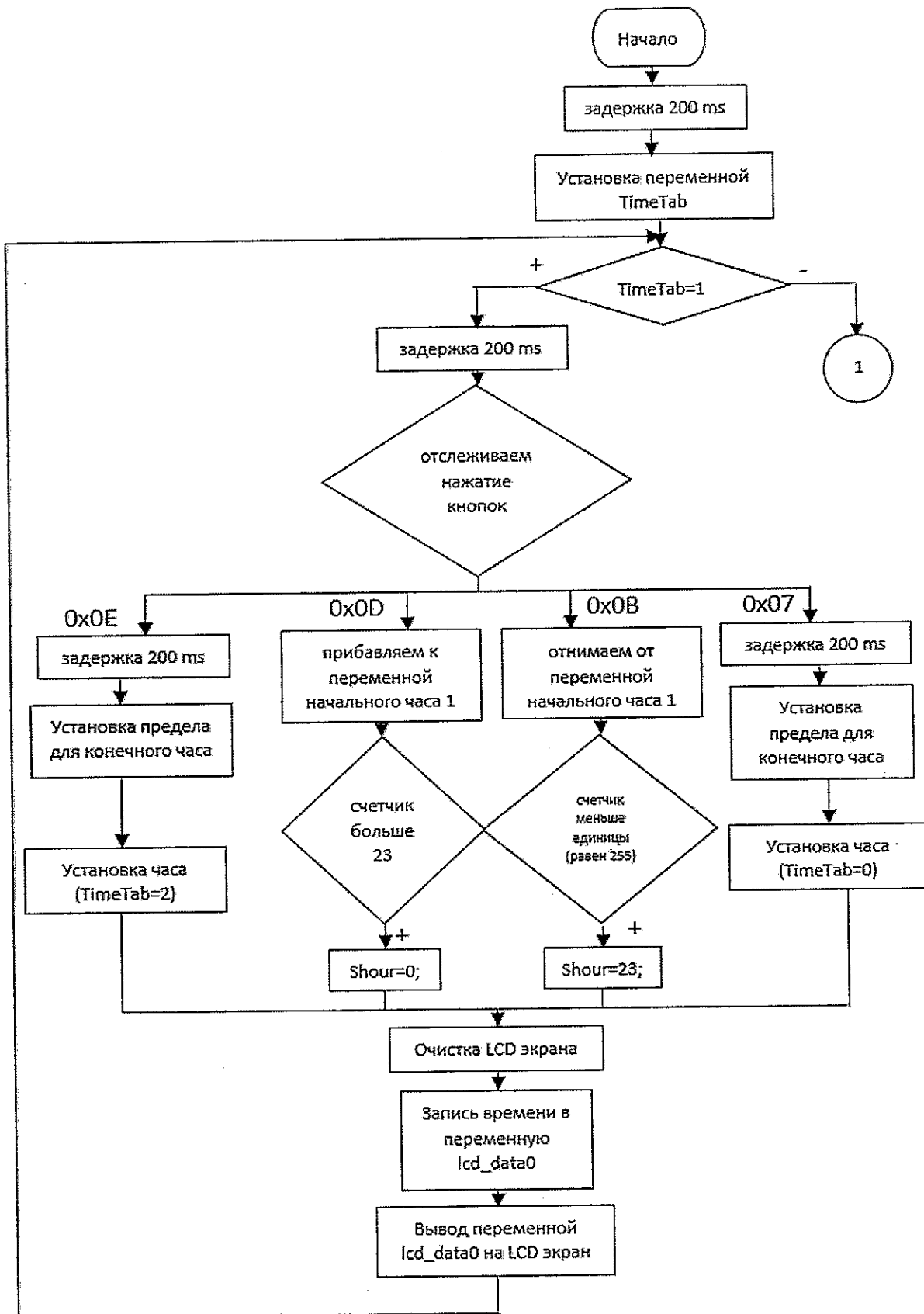


Рисунок А17 - Блок схема void TimerSet(unsigned char count)

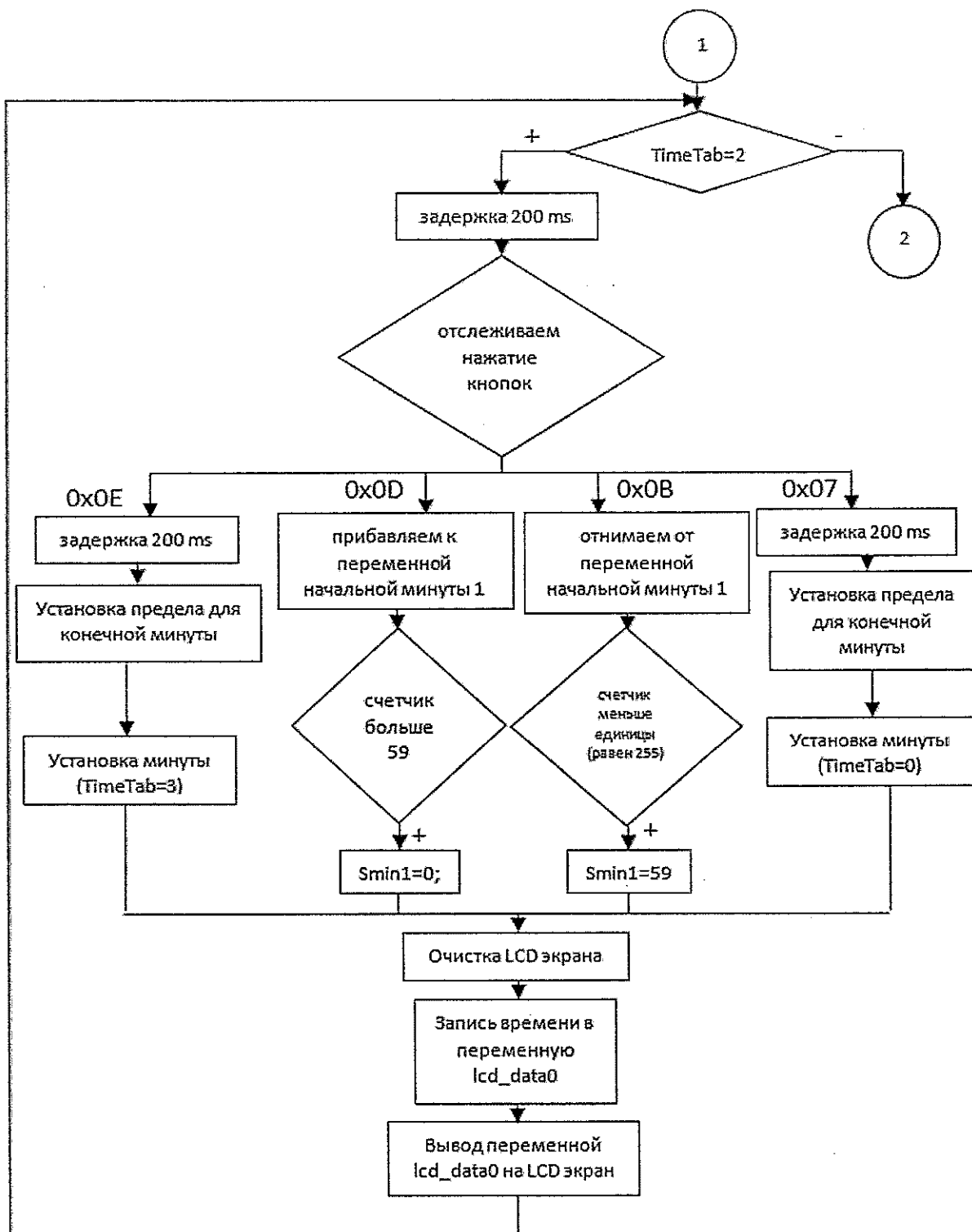


Рисунок А18 - Продолжение рисунка А17

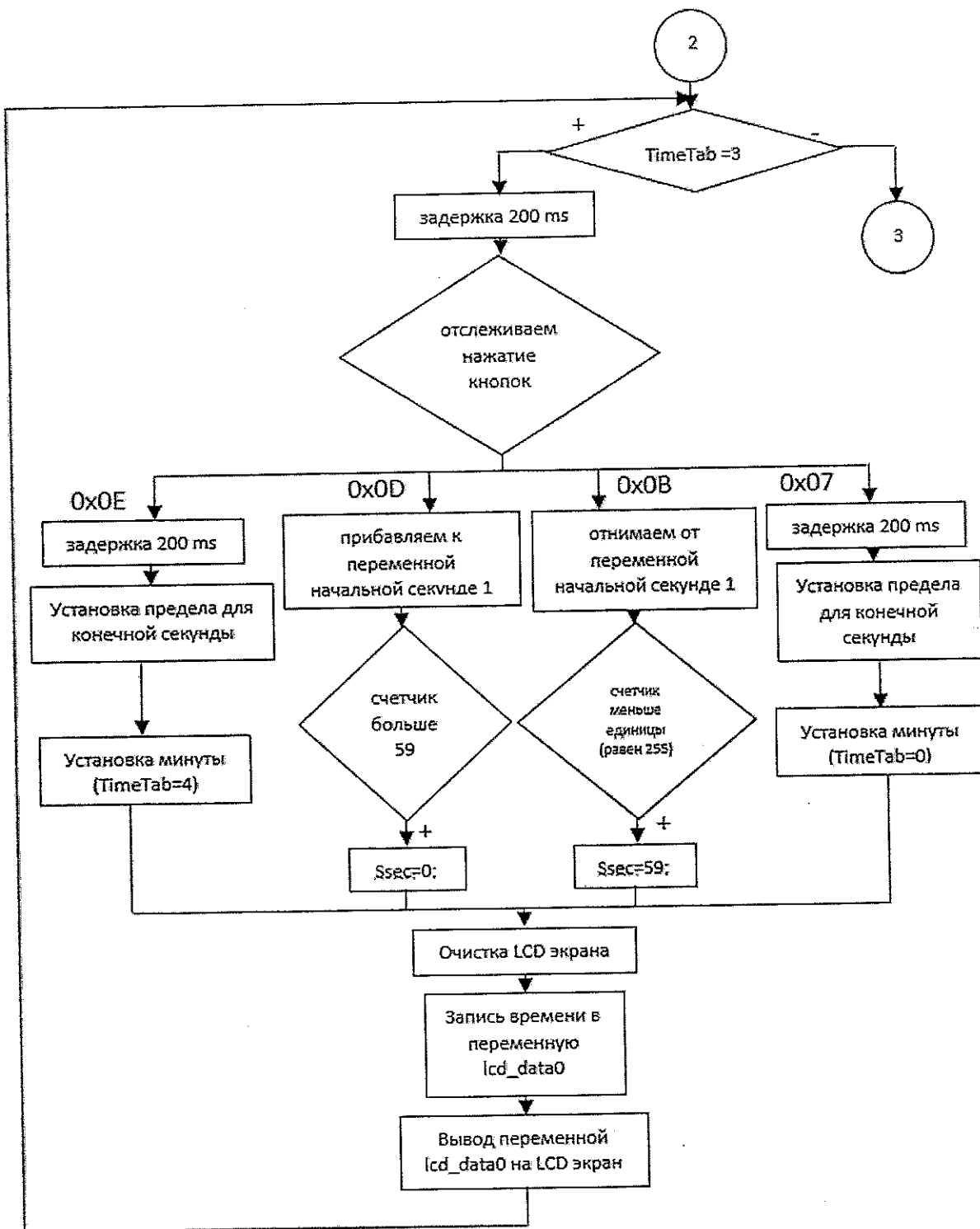


Рисунок А19 - Продолжение рисунка А18

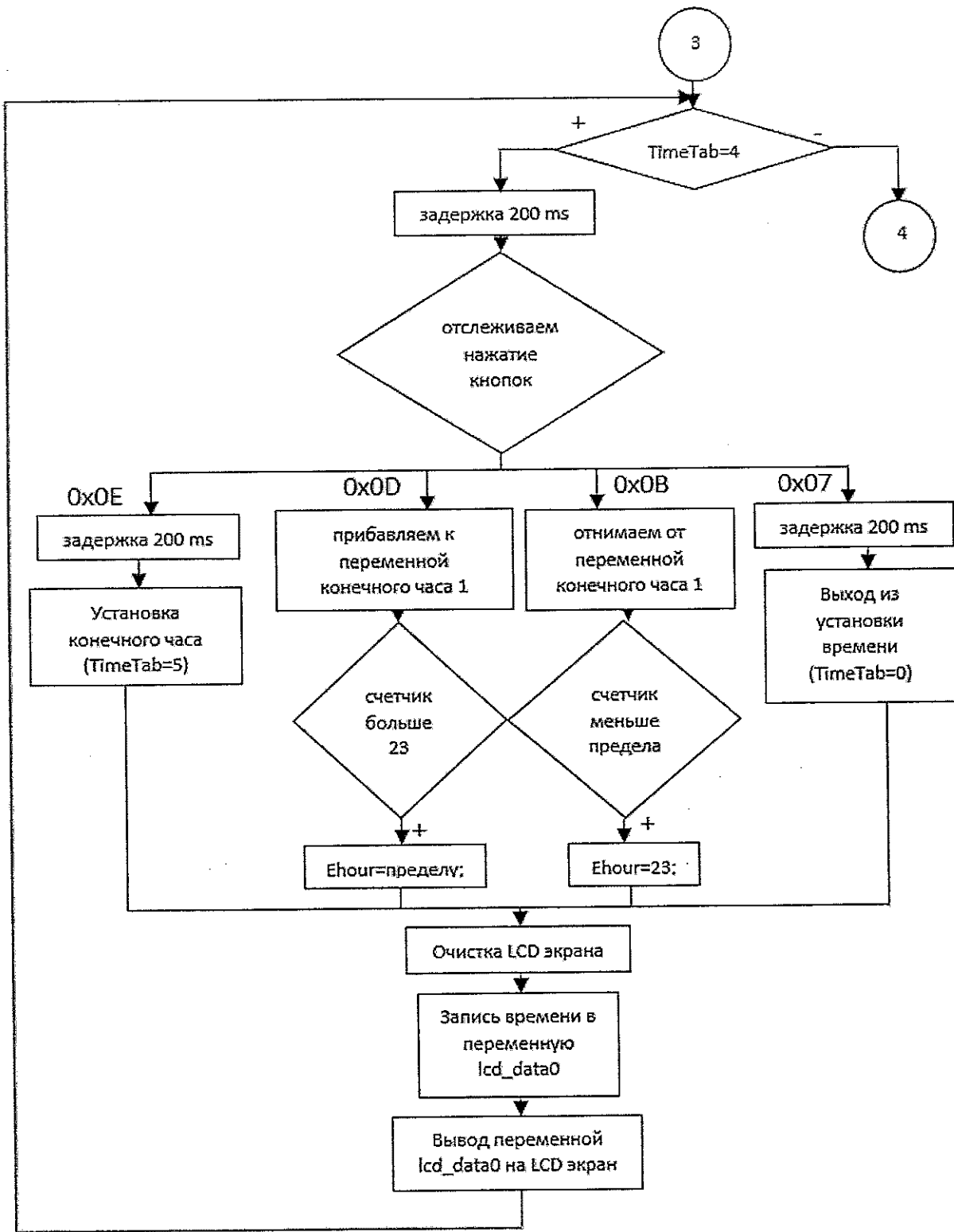


Рисунок А20 - Продолжение рисунка А19

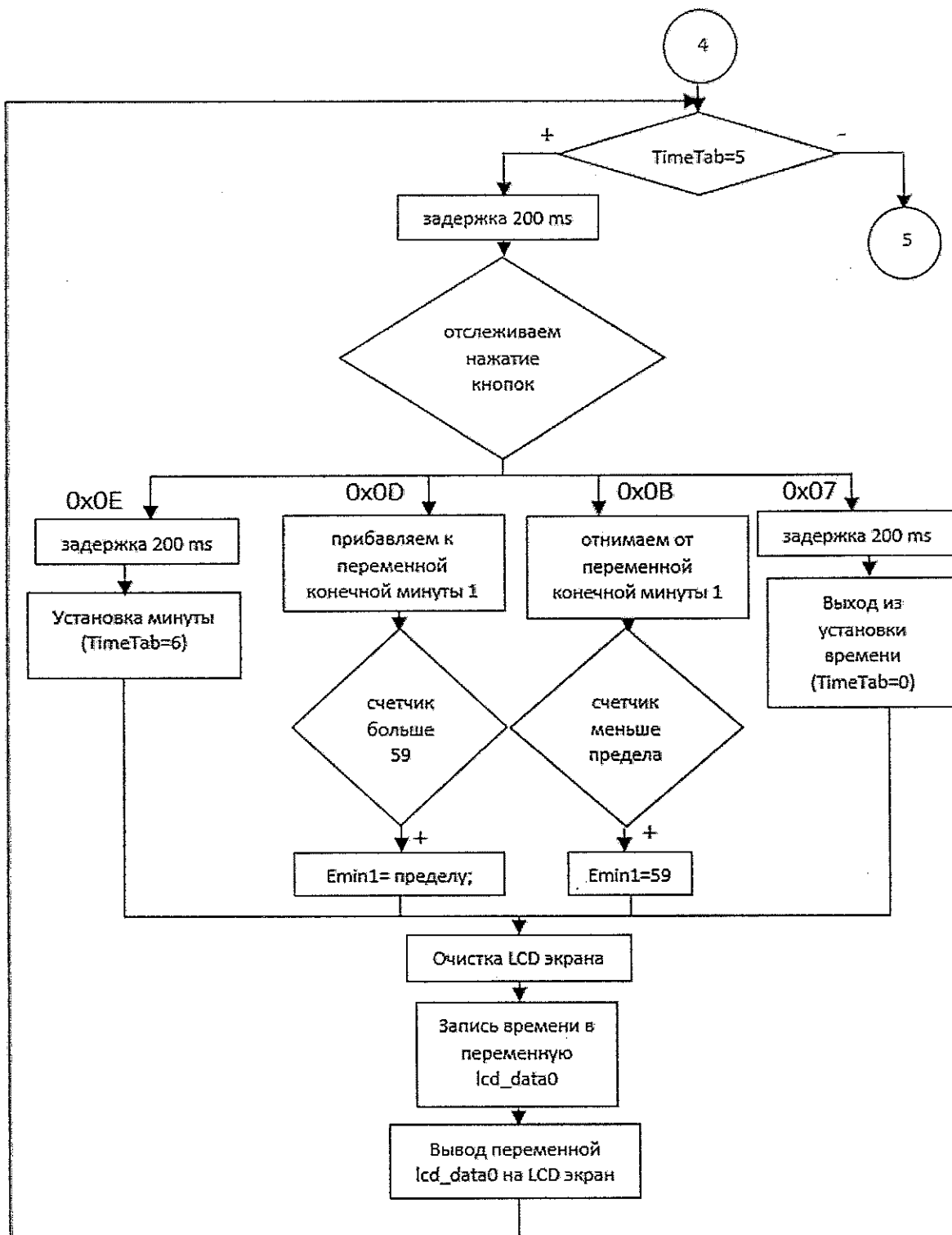


Рисунок А21 - продолжение рисунка А20

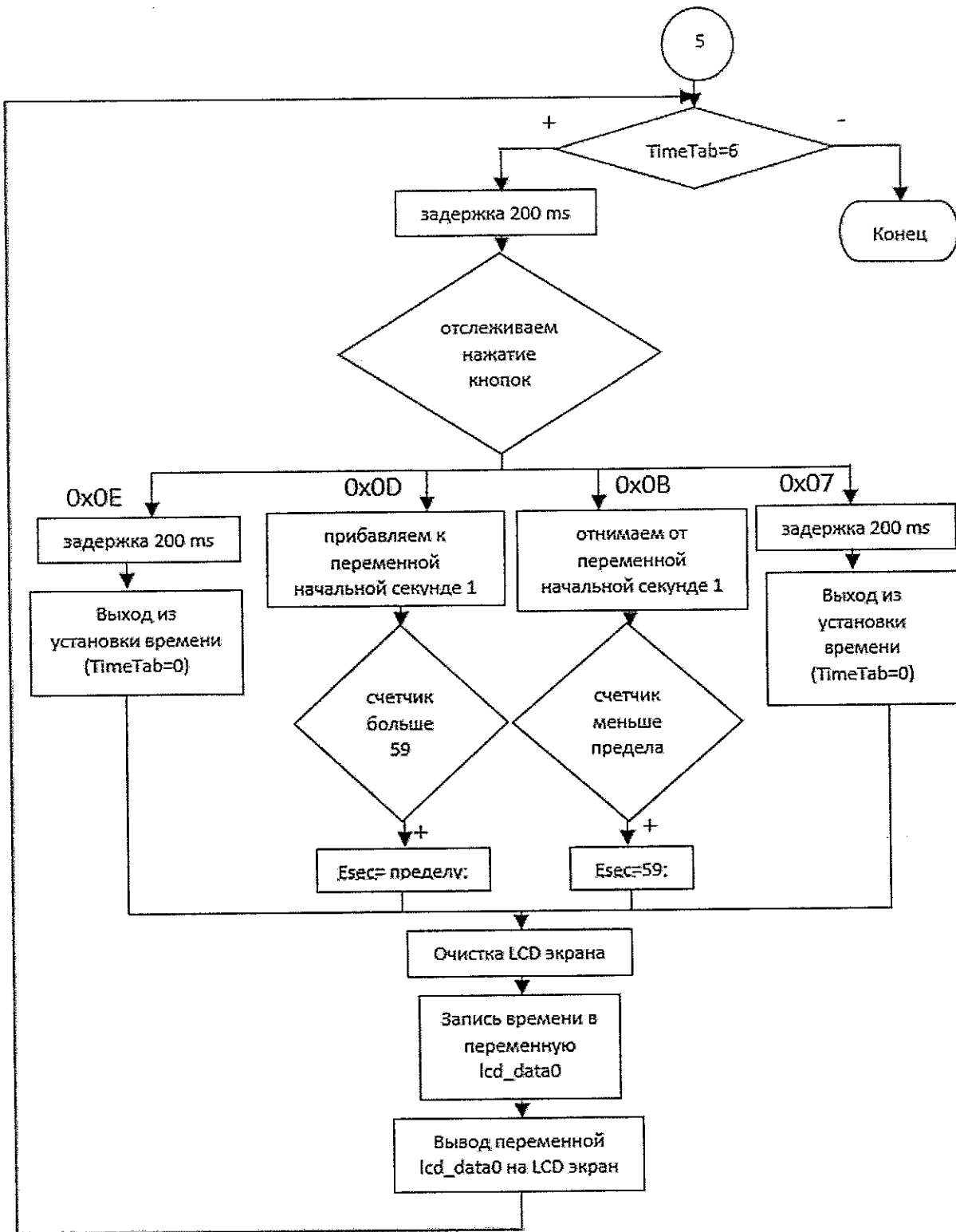


Рисунок А22 - Продолжение рисунка А21

```

Листинг управляющей программы
/*****
//Печь подготовки электродов  Тзад=120(град.С)
Chip type      : ATmega328P
Program type   : Application
AVR Core Clock frequency: 16,000000 MHz
Memory model   : Small
External RAM size : 0
Data Stack size : 512
*****/

#include <mega328p.h>
#include <i2c.h> // I2C Bus functions
#include <ds1307.h> // DS1307 Real Time Clock functions
#include <alcd.h> // Alphanumeric LCD functions
#include <stdio.h> // Standard Input/Output functions
#include <delay.h>

typedef struct
{
    unsigned char Shour;
    unsigned char Smin1;
    unsigned char Ssec;
    unsigned char Ehour;
    unsigned char Emin1;
    unsigned char Esec;
} TIME;
TIME TimeTables[24];

char lcd_data0[20];
char lcd_data1[20];
char lcd_data2[20];
char lcd_data3[20];

unsigned char hour,min1,sec;
unsigned char set=0;
unsigned char TimeTab=0;

void ClokSets();
void TimeTable(void);
void MC_Init(void);
void TimerSet(unsigned char count);
unsigned char TimeTableCheck();
unsigned char TimeCheck(unsigned char current);
void Menu();
int TimeCount();
void TimeTableChangeChase(int v, unsigned char c);
void DeleteTimerTable(unsigned char count);

void main(void)
{
    MC_Init();
    lcd_clear(); //очистка экрана
    while (1)

```



```

{
    Menu();

    PORTD.0 = TimeTableCheck();

    rtc_get_time(&hour,&min1,&sec); //считать время
    sprintf(lcd_data0,"Time: %02i:%02i:%02i  ",hour,min1,sec); // Запись в буфер значения
времени
    lcd_clear(); //очистка экрана
    lcd_puts(lcd_data0); //Выводим на экран текущее время
    delay_ms(200);
};
}

void Menu()
{
    unsigned char m, c = 0;
    if(PINC.0==0)
    {
        m = 1;
        while (m == 1)
        {
            delay_ms(200);
            PORTD.0 = TimeTableCheck();
            switch (PINC&0b00001111) // Отслеживаем нажатие кнопок
            {
                case 0x0E:
                    switch(c)
                    {
                        case 0: ClookSets(); break;
                        case 1: TimeTable(); break;
                    }
                    break; //Если нажата кнопка "Set" то увеличиваем колчество часов на 1
                    case 0x0D: c++; if (c>1){c=0;}; break; //Если нажата кнопка ">" то увеличиваем
колчество часов на 1
                    case 0x0B: c--; if (c==255){c=1;}; break; //Если нажата кнопка "<" то уменьшаем
колчество часов на 1
                    case 0x07: delay_ms(200); m=0; break; //Если нажата кнопка "TimeTable",
переходим к настройкам минут
            };

            if(c == 1)
            {
                sprintf(lcd_data0,"1. Set time ");
                sprintf(lcd_data1, "2.>>Set timetable<<");
            }
            else
            {
                sprintf(lcd_data0,"1.>>Set time<<");
                sprintf(lcd_data1, "2. Set timetable ");
            }

            lcd_clear(); // Очистка экрана
            lcd_puts(lcd_data0); // Вывод на экран текущее значения часов

```

```

        lcd_gotoxy(0,1);
        lcd_puts(lcd_data1);
    }
}

unsigned char TimeTableCheck()
{
    unsigned char ret = 0,i;
    for(i = 0; i <= 23; i++)
    {
        ret = TimeCheck(i);
        if(ret == 1)
            break;
    }
    return ret;
}

unsigned char TimeCheck(unsigned char current)
{
    int curTime = hour*60*60+min1*60+sec;
    int stTime =
TimeTables[current].Shour*60*60+TimeTables[current].Smin1*60+TimeTables[current].Ssec;
    int endTime =
TimeTables[current].Ehour*60*60+TimeTables[current].Emin1*60+TimeTables[current].Esec;
    if(curTime < endTime && curTime >= stTime)
    {
        return 1;
    }
    else return 0;
}

void ClookSets(void)
{
    delay_ms(200);
    set=1;
    while(set==1)
    {
        delay_ms(200);
        switch (PINC&0b00001111) // Отслеживаем нажатие кнопок
        {
            case 0x0D: hour++; if (hour>23){hour=0;}; break; //Если нажата кнопка ">" то
увеличиваем колчество часов на 1
            case 0x0B: hour--; if (hour==255){hour=23;}; break; //Если нажата кнопка "<" то
уменьшаем колчество часов на 1
            case 0x0E: delay_ms(200); set=2; break;
            case 0x07: delay_ms(200); set=0; break;
        };
        lcd_clear(); // Очистка экрана
        sprintf(lcd_data0,"Hour: %02i",hour); //Запись в буфер значения часов
        lcd_puts(lcd_data0); // Вывод на экран текущее значения часов
    };
    while(set==2)
    {

```

```

    delay_ms(200);
    switch (PINC&0b00001111) // Отслеживаем нажатие кнопок
    {
        case 0x0D: min1++; if (min1>59){min1=0;}; break; //Если нажата кнопка ">" то
        увеличиваем колчество минут на 1
        case 0x0B: min1--; if (min1==255){min1=59;}; break; //Если нажата кнопка "<" то
        уменьшаем колчество минут на 1
        case 0x0E: delay_ms(200); set=3; break; //Если нажата кнопка "Set", переходим к
        настройкам минут
        case 0x07: delay_ms(200); set=0; break;
    };
    lcd_clear(); // Очистка экрана
    sprintf(lcd_data0,"Min: %02i",min1); //Запись в буфер значения минут
    lcd_puts(lcd_data0); // Вывод на экран текущее количество минут
};
while(set==3)
{
    delay_ms(200);
    switch (PINC&0b00001111) // Отслеживаем нажатие кнопок
    {
        case 0x0D: sec++; if (sec>59){sec=0;}; break; //Если нажата кнопка ">" то
        увеличиваем колчество минут на 1
        case 0x0B: sec--; if (sec==255){sec=59;}; break; //Если нажата кнопка "<" то
        уменьшаем колчество минут на 1
        case 0x0E: set=0;rtc_set_time(hour,min1,sec); break; //Если нажата кнопка "Set",
        отправляем в микросхему часов новые значения времени
        case 0x07: delay_ms(200); set=0; break;
    };
    lcd_clear(); // Очистка экрана
    sprintf(lcd_data0,"Sec: %02i",sec); //Запись в буфер значения секунд
    lcd_puts(lcd_data0); // Вывод на экран текущее количество секунд
};
}

void TimeTable()
{
    unsigned char i;
    unsigned char count = 0;

    i = 1;
    while (i == 1)
    {
        delay_ms(200);
        PORTD.0 = TimeTableCheck();
        switch (PINC&0b00001111)
        {
            case 0x0E:
                if(count == TimeCount()+1)
                    TimerSet(count);
                else
                    TimeTableChangeChase(TimeCount(), count);
                break;
            case 0x0D: count++; if (count>TimeCount()+1){count=0;}; break;

```

```

        case 0x0B: count--; if (count==255){count=TimeCount()+1;}; break;
        case 0x07: delay_ms(200); l=0; break;
    };

    if(count == TimeCount()+1)
    {
        sprintf(lcd_data0, "Table: %02i/%02i", count + 1, TimeCount()+2);
        sprintf(lcd_data1, "EMPTY");
        sprintf(lcd_data2, ">>ADD NEW<<");
    }
    else
    {
        sprintf(lcd_data0, "Table: %02i/%02i", count + 1, TimeCount()+2);
        sprintf(lcd_data1, "Time start: %02i:%02i:%02i", TimeTables[count].Shour,
TimeTables[count].Smin1, TimeTables[count].Ssec);
        sprintf(lcd_data2, "Time end: %02i:%02i:%02i", TimeTables[count].Ehour,
TimeTables[count].Emin1, TimeTables[count].Esec);
    }

    lcd_clear();
    lcd_puts(lcd_data0);
    lcd_gotoxy(0,1);
    lcd_puts(lcd_data1);
    lcd_gotoxy(0,2);
    lcd_puts(lcd_data2);
}
}

```

```

void TimeTableChangeChase(int v, unsigned char c)
{
    unsigned char l;
    unsigned char count = 0;
    l = 1;
    while (l == 1)
    {
        delay_ms(200);
        PORTD.0 = TimeTableCheck();
        switch (PINC&0b00001111)
        {
            case 0x0E:
                if(count == 0)
                    TimerSet(c);
                else
                    DeleteTimerTable(c);
                l=0;
                break;
            case 0x0D: count++; if (count>1){count=0;}; break;
            case 0x0B: count--; if (count==255){count=0;}; break;
            case 0x07: delay_ms(200); l=0; break;
        };

        sprintf(lcd_data0, "Table: %02i/%02i", c + 1, v+2);
        sprintf(lcd_data1, "Time start: %02i:%02i:%02i", TimeTables[c].Shour,
TimeTables[c].Smin1, TimeTables[c].Ssec);
    }
}

```

```

    sprintf(lcd_data2, "Time end: %02i:%02i:%02i", TimeTables[c].Ehour,
TimeTables[c].Emin1, TimeTables[c].Esec);
    if(count == 0)
        sprintf(lcd_data3, ">>Set<<");
    else
        sprintf(lcd_data3, ">>Delete<<");

    lcd_clear();
    lcd_puts(lcd_data0);
    lcd_gotoxy(0,1);
    lcd_puts(lcd_data1);
    lcd_gotoxy(0,2);
    lcd_puts(lcd_data2);
    lcd_gotoxy(0,3);
    lcd_puts(lcd_data3);
}
}

```

```

void DeleteTimerTable(unsigned char count)

```

```

{
    unsigned char i;

    for(i=count;i<23;i++)
    {
        TimeTables[i].Ehour = TimeTables[i+1].Ehour;
        TimeTables[i].Emin1 = TimeTables[i+1].Emin1;
        TimeTables[i].Esec = TimeTables[i+1].Esec;
        TimeTables[i].Shour = TimeTables[i+1].Shour;
        TimeTables[i].Smin1 = TimeTables[i+1].Smin1;
        TimeTables[i].Ssec = TimeTables[i+1].Ssec;
    }

    TimeTables[23].Ehour = 0;
    TimeTables[23].Emin1 = 0;
    TimeTables[23].Esec = 0;
    TimeTables[23].Shour = 0;
    TimeTables[23].Smin1 = 0;
    TimeTables[23].Ssec = 0;
}

```

```

int TimeCount()

```

```

{
    int time, v = -1;
    unsigned char c;

    for(c=0;c<24;c++)
    {
        time = TimeTables[c].Shour*60*60+TimeTables[c].Smin1*60+TimeTables[c].Ssec+
            TimeTables[c].Ehour*60*60+TimeTables[c].Emin1*60+TimeTables[c].Esec;
        if(time != 0)
            v = c;
    }

    return v;
}

```

```

}

void TimerSet(unsigned char count)
{
    TimeTab=1;
    while(TimeTab==1)
    {
        delay_ms(200);
        PORTD.0 = TimeTableCheck();
        switch (PINC&0b00001111) // Отслеживаем нажатие кнопок
        {
            case 0x07: delay_ms(200); TimeTables[count].Ehour=TimeTables[count].Shour;
            TimeTab=0; break;
            case 0x0D: TimeTables[count].Shour++; if
            (TimeTables[count].Shour>23){TimeTables[count].Shour=0;}; break; //Если нажата кнопка
            ">" то увеличиваем колчество часов на 1
            case 0x0B: TimeTables[count].Shour--; if
            (TimeTables[count].Shour==255){TimeTables[count].Shour=23;}; break; //Если нажата
            кнопка "<" то уменьшаем колчество часов на 1
            case 0x0E: delay_ms(200); TimeTables[count].Ehour=TimeTables[count].Shour;
            TimeTab=2; break; //Если нажата кнопка "TimeTable", переходим к настройкам минут
        };
        lcd_clear(); // Очистка экрана
        sprintf(lcd_data0,"Start hour: %02i",TimeTables[count].Shour); //Запись в буфер
        значения часов
        lcd_puts(lcd_data0); // Вывод на экран текущее значения часов
    };
    while(TimeTab==2)
    {
        delay_ms(200);
        PORTD.0 = TimeTableCheck();
        switch (PINC&0b00001111) // Отслеживаем нажатие кнопок
        {
            case 0x07: delay_ms(200); TimeTables[count].Emin1=TimeTables[count].Smin1;
            TimeTab=0; break;
            case 0x0D: TimeTables[count].Smin1++; if
            (TimeTables[count].Smin1>59){TimeTables[count].Smin1=0;}; break; //Если нажата кнопка
            ">" то увеличиваем колчество минут на 1
            case 0x0B: TimeTables[count].Smin1--; if
            (TimeTables[count].Smin1==255){TimeTables[count].Smin1=59;}; break; //Если нажата
            кнопка "<" то уменьшаем колчество минут на 1
            case 0x0E: delay_ms(200); TimeTables[count].Emin1=TimeTables[count].Smin1;
            TimeTab=3; break; //Если нажата кнопка "TimeTable", переходим к настройкам секунд
        };
        lcd_clear(); // Очистка экрана
        sprintf(lcd_data0,"Start min: %02i",TimeTables[count].Smin1); //Запись в буфер
        значения минут
        lcd_puts(lcd_data0); // Вывод на экран текущее количество минут
    };
    while(TimeTab==3)
    {
        delay_ms(200);
        PORTD.0 = TimeTableCheck();
        switch (PINC&0b00001111) // Отслеживаем нажатие кнопок

```

```

    {
        case 0x07: delay_ms(200); TimeTables[count].Esec=TimeTables[count].Ssec + 1;
        TimeTab=0; break;
        case 0x0D: TimeTables[count].Ssec++; if
        (TimeTables[count].Ssec>59){TimeTables[count].Ssec=0;}; break; //Если нажата кнопка ">"
        то увеличиваем колчество минут на 1
        case 0x0B: TimeTables[count].Ssec--; if
        (TimeTables[count].Ssec==255){TimeTables[count].Ssec=59;}; break; //Если нажата кнопка
        "<" то уменьшаем колчество минут на 1
        case 0x0E: delay_ms(200); TimeTables[count].Esec=TimeTables[count].Ssec + 1;
        TimeTab=4; break; //Если нажата кнопка "TimeTable", переходим к настройке колчества
        минут задержки до выкл.
    };

    lcd_clear(); // Очистка экрана
    sprintf(lcd_data0,"Start sec: %02i",TimeTables[count].Ssec); //Запись в буфер значения
    секунд
    lcd_puts(lcd_data0); // Вывод на экран текущее количество секунд
};
while(TimeTab==4)
{
    delay_ms(200);
    PORTD.0 = TimeTableCheck();
    switch (PINC&0b00001111) // Отслеживаем нажатие кнопок
    {
        case 0x07: delay_ms(200); TimeTab=0; break;
        case 0x0D: TimeTables[count].Ehour++; if
        (TimeTables[count].Ehour>23){TimeTables[count].Ehour=TimeTables[count].Shour;}; break;
        //Если нажата кнопка ">" то увеличиваем колчество часов на 1
        case 0x0B: TimeTables[count].Ehour--; if (TimeTables[count].Ehour==255 ||
        TimeTables[count].Ehour<TimeTables[count].Shour){TimeTables[count].Ehour=23;}; break;
        //Если нажата кнопка "<" то уменьшаем колчество часов на 1
        case 0x0E: delay_ms(200); TimeTab=5; break; //Если нажата кнопка "TimeTable",
        переходим к настройкам минут
    };
    lcd_clear(); // Очистка экрана
    sprintf(lcd_data0,"End hour: %02i",TimeTables[count].Ehour); //Запись в буфер
    значения часов
    lcd_puts(lcd_data0); // Вывод на экран текущее значения часов
};
while(TimeTab==5)
{
    delay_ms(200);
    PORTD.0 = TimeTableCheck();
    switch (PINC&0b00001111) // Отслеживаем нажатие кнопок
    {
        case 0x07: delay_ms(200); TimeTab=0; break;
        case 0x0D: TimeTables[count].Emin1++;
        if (TimeTables[count].Emin1>59)
        {
            if(TimeTables[count].Ehour == TimeTables[count].Shour)
            TimeTables[count].Emin1=TimeTables[count].Smin1;
            else
            TimeTables[count].Emin1=0;
        }
    }
}

```

```

    }
    break; //Если нажата кнопка ">" то увеличиваем колчество минут на 1
case 0x0B: TimeTables[count].Emin1--;
    if(TimeTables[count].Ehour == TimeTables[count].Shour)
    {
        if (TimeTables[count].Emin1==255 ||
TimeTables[count].Emin1<TimeTables[count].Smin1)
            TimeTables[count].Emin1=59;
        }
    else
        if (TimeTables[count].Emin1==255)
            TimeTables[count].Emin1=59;
        break; //Если нажата кнопка "<" то уменьшаем колчество минут на 1
case 0x0E: delay_ms(200); TimeTab=6; break; //Если нажата кнопка "TimeTable",
переходим к настройкам секунд
    };
    lcd_clear(); // Очистка экрана
    sprintf(lcd_data0,"End min: %02i",TimeTables[count].Emin1); //Запись в буфер
значения минут
    lcd_puts(lcd_data0); // Вывод на экран текущее количество минут
    };
    while(TimeTab==6)
    {
        delay_ms(200);
        PORTD.0 = TimeTableCheck();
        switch (PINC&0b00001111) // Отслеживаем нажатие кнопок
        {
            case 0x07: delay_ms(200); TimeTab=0; break;
            case 0x0D: TimeTables[count].Esec++;
                if (TimeTables[count].Esec>59)
                {
                    if(TimeTables[count].Ehour == TimeTables[count].Shour &&
TimeTables[count].Emin1 == TimeTables[count].Smin1)
                        TimeTables[count].Esec=TimeTables[count].Ssec + 1;
                    else
                        TimeTables[count].Esec=0;
                }
                break; //Если нажата кнопка ">" то увеличиваем колчество минут на 1
            case 0x0B: TimeTables[count].Esec--;
                if(TimeTables[count].Ehour == TimeTables[count].Shour &&
TimeTables[count].Emin1 == TimeTables[count].Smin1)
                {
                    if (TimeTables[count].Esec==255 ||
TimeTables[count].Esec<TimeTables[count].Ssec + 1)
                        TimeTables[count].Esec=59;
                }
            else
                if (TimeTables[count].Esec==255)
                    TimeTables[count].Esec=59;
                break; //Если нажата кнопка "<" то уменьшаем колчество минут на 1
            case 0x0E: delay_ms(200); TimeTab=4; break; //Если нажата кнопка "TimeTable",
переходим к настройке колчества минут задержки до выкл.
        }
    };

```



```

    lcd_clear(); // Очистка экрана
    sprintf(lcd_data0,"End sec: %02i",TimeTables[count].Esec); //Запись в буфер значения
секунд
    lcd_puts(lcd_data0); // Вывод на экран текущее количество секунд
};
}

```

```

void MC_Init(void)

```

```

{
#pragma optsize-
CLKPR=(1<<CLKPCE);
CLKPR=(0<<CLKPCE) | (0<<CLKPS3) | (0<<CLKPS2) | (0<<CLKPS1) | (0<<CLKPS0);
#ifdef _OPTIMIZE_SIZE_
#pragma optsize+
#endif
 DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) | (0<<DDB2) |
(0<<DDB1) | (0<<DDB0);
 PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) |
(0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);
 DDRC=(0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) |
(0<<DDC0);
 PORTC=(0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) |
(0<<PORTC1) | (0<<PORTC0);
 DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) |
(0<<DDD1) | (1<<DDD0);
 PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) |
(0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);

 TCCR0A=(0<<COM0A1) | (0<<COM0A0) | (0<<COM0B1) | (0<<COM0B0) | (0<<WGM01) |
(0<<WGM00);
 TCCR0B=(0<<WGM02) | (0<<CS02) | (0<<CS01) | (0<<CS00);
 TCNT0=0x00;
 OCR0A=0x00;
 OCR0B=0x00;

 TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) |
(0<<WGM10);
 TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11)
| (0<<CS10);
 TCNT1H=0x00;
 TCNT1L=0x00;
 ICR1H=0x00;
 ICR1L=0x00;
 OCR1AH=0x00;
 OCR1AL=0x00;
 OCR1BH=0x00;
 OCR1BL=0x00;

 ASSR=(0<<EXCLK) | (0<<AS2);
 TCCR2A=(0<<COM2A1) | (0<<COM2A0) | (0<<COM2B1) | (0<<COM2B0) | (0<<WGM21) |
(0<<WGM20);
 TCCR2B=(0<<WGM22) | (0<<CS22) | (0<<CS21) | (0<<CS20);
 TCNT2=0x00;
 OCR2A=0x00;

```

OCR2B=0x00;

TIMSK0=(0<<OCIE0B) | (0<<OCIE0A) | (0<<TOIE0);

TIMSK1=(0<<ICIE1) | (0<<OCIE1B) | (0<<OCIE1A) | (0<<TOIE1);

TIMSK2=(0<<OCIE2B) | (0<<OCIE2A) | (0<<TOIE2);

EICRA=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);

EIMSK=(0<<INT1) | (0<<INT0);

PCICR=(0<<PCIE2) | (0<<PCIE1) | (0<<PCIE0);

UCSR0A=(0<<RXC0) | (0<<TXC0) | (0<<UDRE0) | (0<<FE0) | (0<<DOR0) | (0<<UPE0) |
(0<<U2X0) | (0<<MPCM0);

UCSR0B=(0<<RXCIE0) | (0<<TXCIE0) | (0<<UDRIE0) | (1<<RXEN0) | (1<<TXEN0) |
(0<<UCSZ02) | (0<<RXB80) | (0<<TXB80);

UCSR0C=(0<<UMSEL01) | (0<<UMSEL00) | (0<<UPM01) | (0<<UPM00) | (0<<USBS0) |
(1<<UCSZ01) | (1<<UCSZ00) | (0<<UCPOL0);

UBRR0H=0x00;

UBRR0L=0x67;

ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) |
(0<<ACIS1) | (0<<ACIS0);

ADCSRB=(0<<ACME);

DIDR1=(0<<AIN0D) | (0<<AIN1D);

ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) |
(0<<ADPS1) | (0<<ADPS0);

SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) |
(0<<SPR1) | (0<<SPR0);

TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

i2c_init();

rtc_init(0,0,0);

lcd_init(20);


}

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

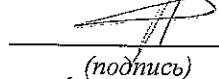
СОГЛАСОВАНО

Начальник отдела ОНиПКРС

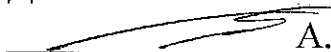

(подпись) Е.М. Димитриади
« 19 » 06 20 22 г.

УТВЕРЖДАЮ

Проректор по научной работе,
д-р техн. наук, профессор


(подпись) А.В. Космынин
« 19 » 06 20 22 г.

Декан ФЭУ


(подпись) А.С. Гудим
« 19 » 06 20 22 г.

АКТ

о приемке в эксплуатацию проекта

Аппаратно-программный комплекс

«Контроллер для управления силовой нагрузкой по расписанию»

г. Комсомольск-на-Амуре

« 20 » 12 2022 г.

Комиссия в составе представителей:

со стороны заказчика

- С.И. Сухоруков – руководитель СКБ,
- А.С. Гудим – декан ФЭУ

со стороны исполнителя

- В.А. Егоров – руководитель проекта,
- Д. А. Логинов – ОМРб-1

составила акт о нижеследующем:

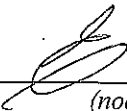
«Исполнитель» передает проект «Контроллер для управления силовой нагрузкой по расписанию», в составе:

1. Блок управления силовой нагрузкой по расписанию.

2. Рабочие программы управления изделием.

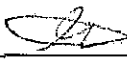
3. Паспорт изделия

Руководитель проекта

 20.12.2022
(подпись, дата)


В.А. Егоров

Исполнитель проекта


(подпись, дата)

Д. А. Логинов

Таблица учета проектной работы в учебных дисциплинах

Дисциплина	Форма учтенной работы (номер ЛР, КП, КР, РГЗ, зачет, итоговая оценка, экзамен)	Преподаватель (дата, ФИО, подпись)	Примечание (ЗУН полученные при выполнении проекта)
Микропроцессорные устройства систем управления	КП	28.12.22 В.А.Егоров 	31(ПК-6-2) Правила составления структуры и алгоритма работы микроконтроллера на различных стадиях проектирования системы электропривода 32(ПК-6-2) Типовые решения по структуре и алгоритмам работы микропроцессорной системы электропривода У1(ПК-6-2) Составлять алгоритмы работы микроконтроллера на различных стадиях проектирования системы электропривода У2(ПК-6-2) Осуществлять сбор и обработку справочной информации по типовым решениям о структуре и алгоритме работы микропроцессорной системы электропривода Н1(ПК-6-2) Анализ технического задания на составление алгоритма работы при проектировании микропроцессорной системы электропривода Н2(ПК-6-2) Выбор оптимальных технических решений по структуре и алгоритму работы микропроцессорной системы электропривода