


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»

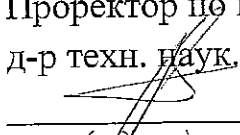
СОГЛАСОВАНО

Начальник отдела ОНИПКРС


  
(подпись) Е.М. Димитриади  
« 19 » 06 20 23 г.

УТВЕРЖДАЮ

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

  
(подпись) А.В. Космынин  
« 19 » 06 20 23 г.

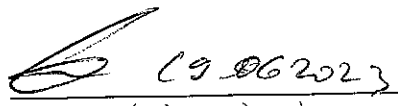
Декан ФЭУ

  
(подпись) А.С. Гудим  
« 19 » 06 20 23 г.

Аппаратно-программный комплекс  
«Микроконтроллерная система капельного полива растений»

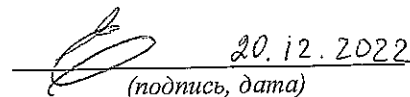
Комплект конструкторской документации

Руководитель СКБ

  
(подпись, дата)

С.И. Сухоруков

Руководитель проекта

  
(подпись, дата)

В.А. Егоров

Комсомольск-на-Амуре 2022

## Карточка проекта

<b>Название</b>	Микроконтроллерная система капельного полива растений
<b>Тип проекта</b>	Тип проекта: учебная работа
<b>Исполнители</b>	И.А. Иванова, М.А. Едич
<b>Срок реализации</b>	14.09.2022- 31.12.2022

### Использованные материалы и компоненты

QTY	PART-REFS	VALUE	CODE
-----			
Resistors			
-----			
5	R1-R5	4.7K	Digikey 311-4.7KETR-ND
1	R6	1K	Digikey 311-4.7KETR-ND
Integrated Circuits			
-----			
1	U1	ATMEGA328P_32PIN	
1	U2	DS1307	
Diodes			
-----			
1	D1	LED-GREEN	
Transistors			
-----			
1	Q1	2N2222	
Relays			
-----			
1	RL1	QUAZ-SH-105D	
Miscellaneous			
-----			
1	BAT1	3V	
1	LCD1	LM044L	
1	RV1	10k	
1	X1	32768Hz	
3	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: Ивановой И.А., Едичу М.А. группы 0МР6-1

Название проекта: Микроконтроллерная система капельного полива растений

Назначение: Автоматизация поливных работ в теплице

Область использования: Бытовое применение

Функциональное описание устройства:

Система осуществляет управление поливом тепличных растений по расписанию.

Техническое описание устройства: Блок управления системы капельного полива, содержащий: однокристалльный микроконтроллер; интерфейс для управления исполнительным оборудованием; жидкокристаллический алфавитно-цифровой индикатор для отображения даты, времени и расписания полива; кнопки ввода даты, времени и расписания полива; блок питания с выходным напряжением +5В.

Требования: длительность расписания -24 часа.

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:

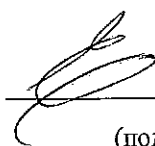
---

---

Перечень графического материала:

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

Руководитель проекта



29.12.2022

(подпись, дата)

В.А. Егоров

Министерство науки и высшего образования Российской Федерации

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

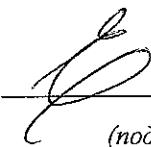


## ПАСПОРТ

### Аппаратно-программный комплекс

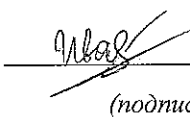
«Микроконтроллерная система капельного полива растений»

Руководитель проекта

29.12.2022  
(подпись, дата)

*В.А. Егоров*

Ответственный исполнитель

29.12.2022  
(подпись, дата)

*И.А. Иванова*

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
5.3	Порядок настройки режимов работы изделия .....	13
6	Условия эксплуатации .....	17
6.1	Правила и особенности размещения изделия .....	17
6.2	Меры безопасности .....	18
6.3	Правила хранения и транспортирования .....	18

## **1 Общие положения**

Настоящий паспорт является документом, предназначенным для ознакомления с основными техническими характеристиками, устройством, правилами установки и эксплуатации устройства «Микроконтроллерная система капельного полива растений» (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 *«Микроконтроллерная система капельного полива растений»* (АПК МКСКПР).

### **1.2 Наименования документов, на основании которых ведется проектирование системы**

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

- задание на разработку.

### **1.3 Перечень организаций, участвующих в разработке системы**

Заказчиком создания АПК МКСКПР 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 МКСКПР являются студенты группы 0МР6-1 И.А. Иванова, М.А. Едич

#### **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 МКСКПР – представляет из себя устройство полива расписаний в теплице по расписанию.

В состав изделия входят: однокристалльный микроконтроллер; интерфейс для управления исполнительным оборудованием; жидкокристаллический алфавитно-цифровой индикатор для отображения даты, времени и расписания полива; кнопки ввода даты, времени и расписания полива; блок питания с выходным напряжением +5В.

### **2.2 Области использования изделия**

Изделие может использоваться для полива растений по расписанию.

### **2.3 Принцип действия**

Пользователь включает установку. Вводит расписание включения/выключения подачи воды для полива растений. Установка включает/выключает подачу воды 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	84
Диапазон рабочих температур прибора, С°	-55 ... +125
Длительность интервала расписания	24 часа
Количество каналов управления	1
Питание, В	5
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 полива растений.

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 потока воды;

Блок коммутации силовой нагрузки представляет из себя нормально закрытый электромагнитный клапан.

Алфавитно-цифровой индикатор – служит для контроля коррекции времени и ввода расписания полива растений.

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 подачи воды для полива растений. Установка включает/выключает подачу воды по расписанию.

Блок-схема работы управляющей программы приведена в Приложении А.

## 5.3 Порядок настройки режимов работы изделия

Во время первого запуска на экране отображаются исходные дата и время (рисунок 1).

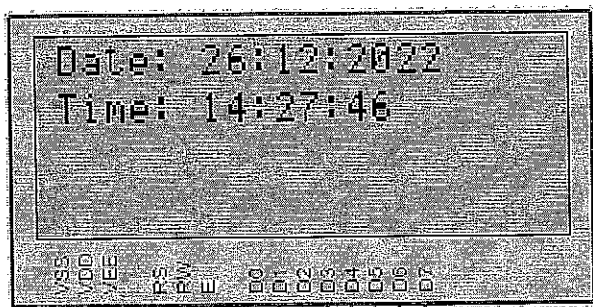


Рисунок 1 – Исходное состояние экрана (отображены текущие дата и время)

Чтобы произвести настройку времени или расписания, нужно нажать

кнопку set, тогда появится меню для настройки, где будут 3 пункта: калибровка времени, настройка таблицы расписаний и выход на первоначальный экран (рисунок 2).

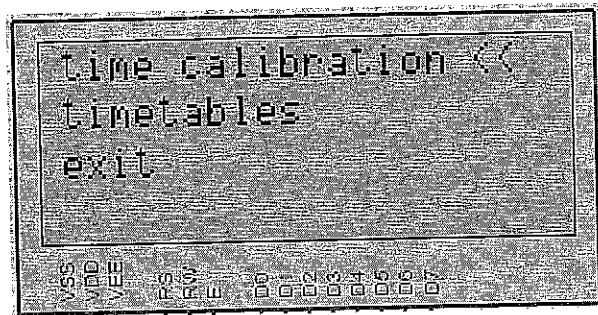
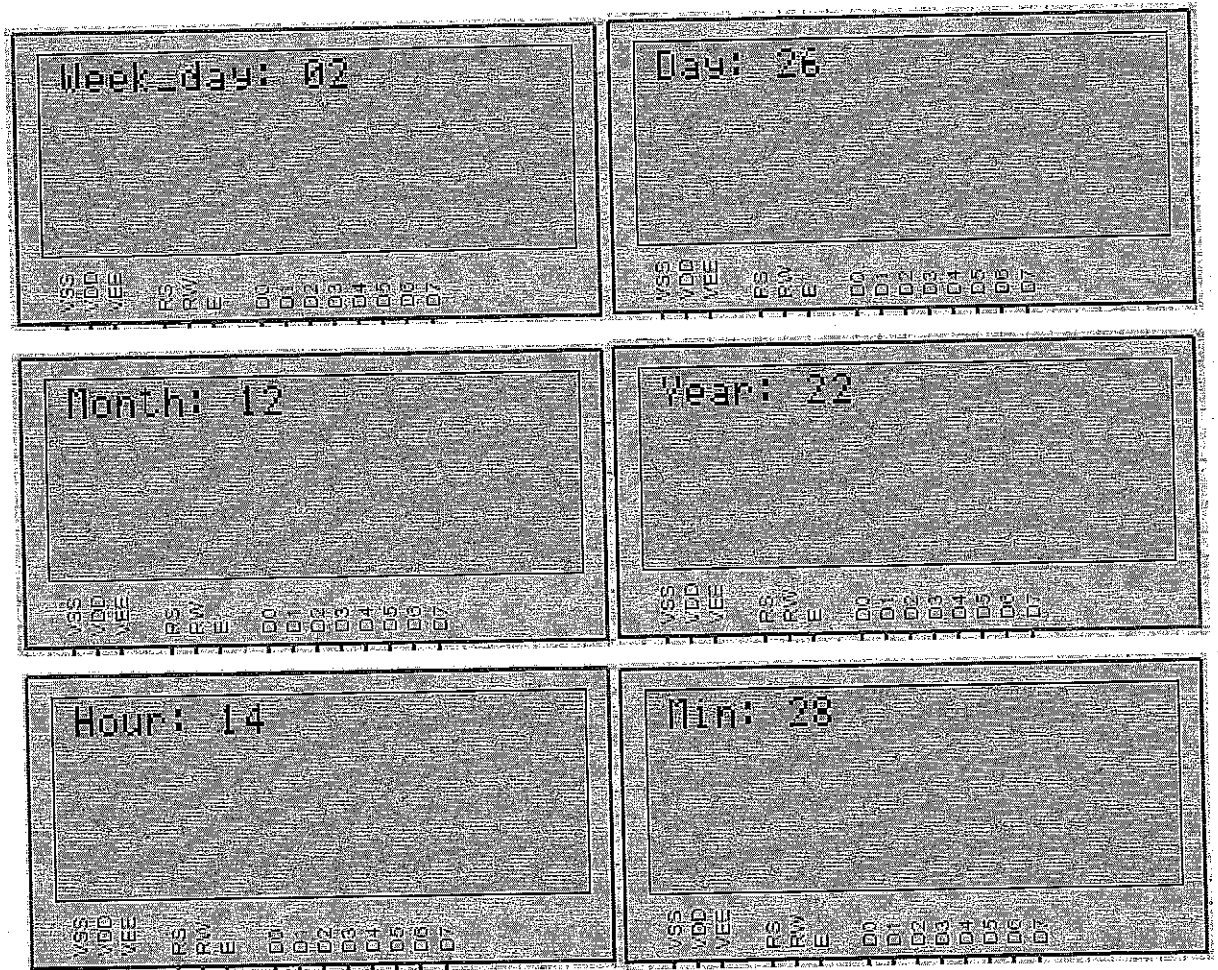


Рисунок 2 – выбор настройки времени в главном меню

При выборе пункта калибровка времени можно поэтапно произвести настройку даты и времени от текущей недели до секунд.



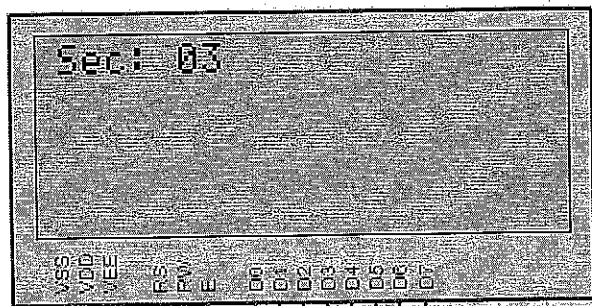


Рисунок 3 – Установка даты и времени

После калибровки времени состояние экрана станет исходным (рисунок 1).

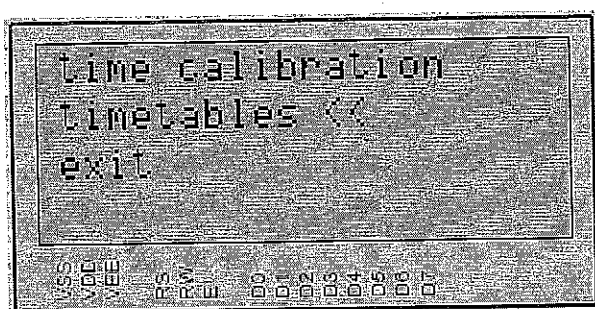


Рисунок 4 – выбор настройки расписания в главном меню

При выборе в том же меню пункта настройки расписания (рисунок 4) появляется список расписаний (рисунок 5), с номерами от #00 до #09. Где можно выбрать любое расписание для настройки и также как и с калибровкой поэтапно настроить стартовое время и продолжительности работы (рисунок 6).

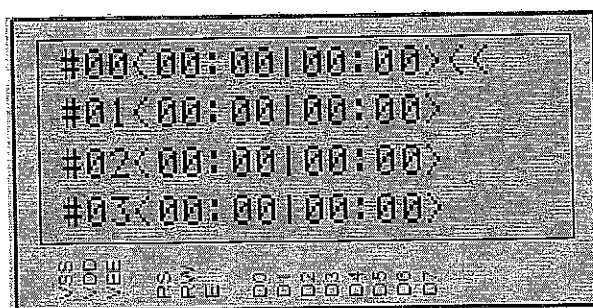


Рисунок 5 –таблица расписаний

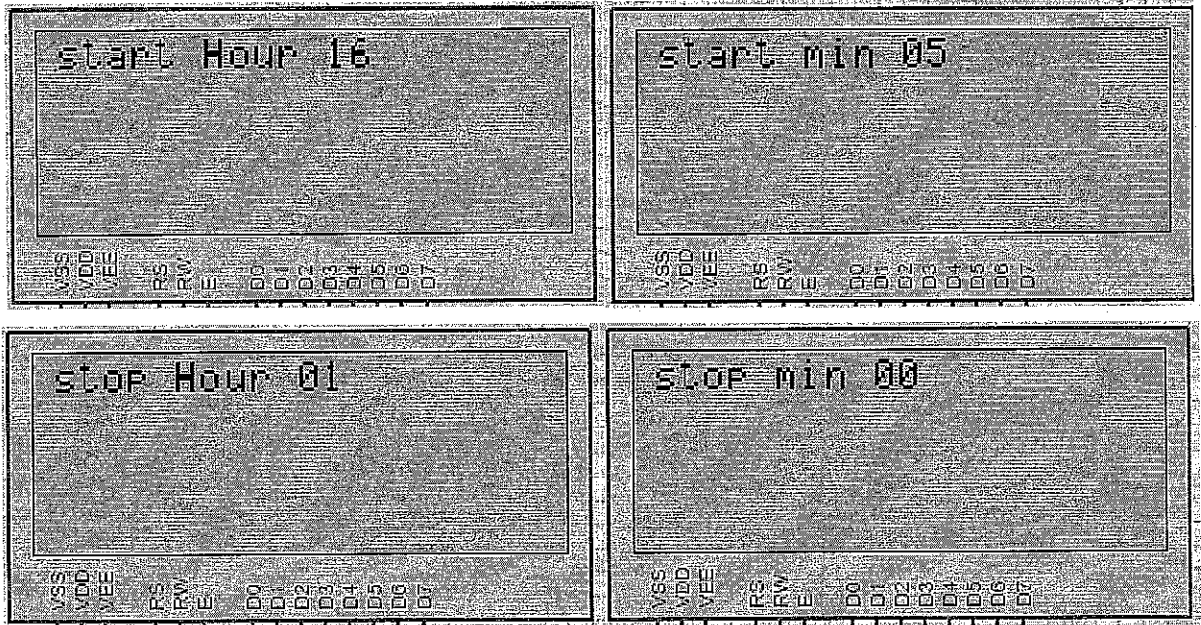


Рисунок 6 –настройка первого расписания

Результат настройки нулевого расписания #00 показано на рисунке 7.

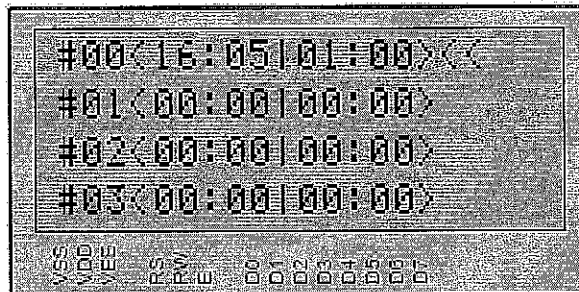


Рисунок 7 – Измененная таблица расписаний



## **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  $20 \pm 5$  °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.



Спецификация к принципиальной схеме

Поз. обо- знач.	Наименование	Кол.	Примечание
	Транзисторы		
Q1	2N2222	1	
	Светодиоды		
D1	LED-GREEN	1	
	Резисторы		
R1-R5	4.7K	5	
R6	1K	1	
	Интегральные схемы		
U2	DS1307	1	
	Модули		
M1	ARDUINO_NANO	1	
	Реле		
RL1	QUAZ-SH-105D	1	
	Разное		
BAT1	3V	1	
LCD1	LM044L	1	
X1	32768Hz	1	
RV1	10K	1	Digikey 3252W- 203LF-ND
key		3	

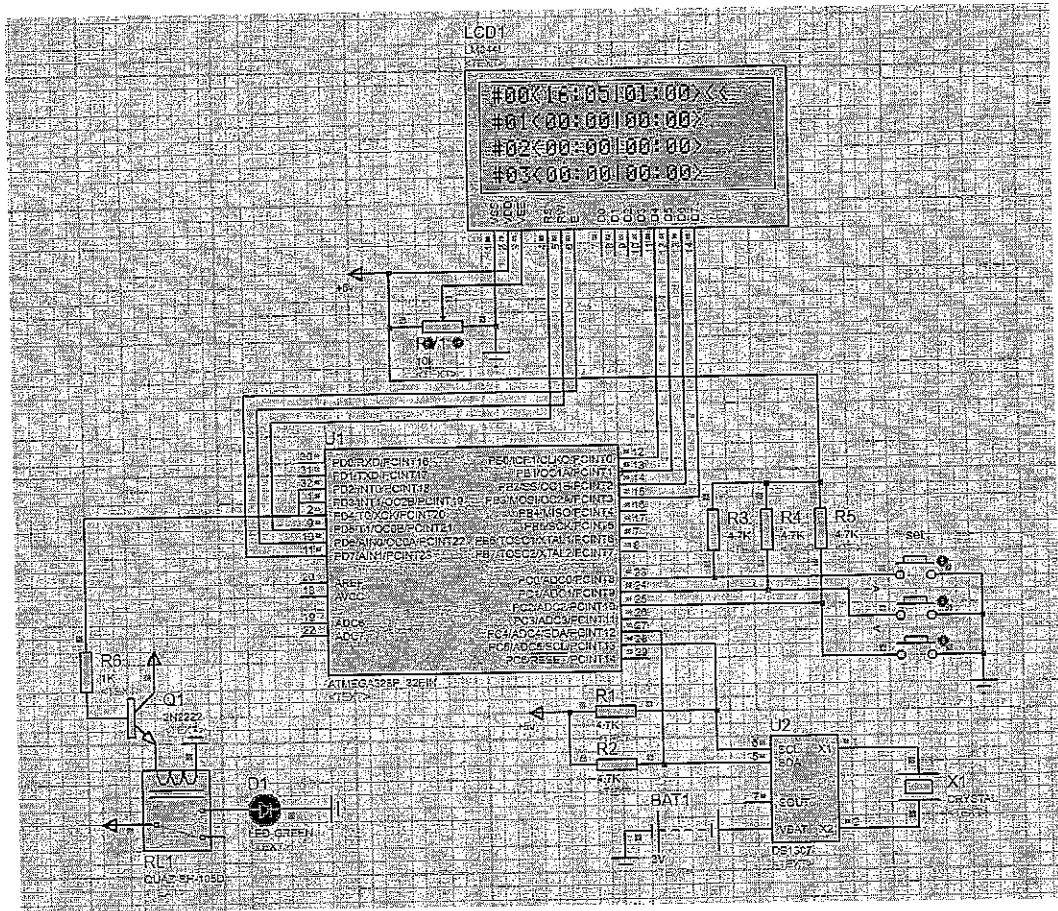


Рисунок А2 – Модель изделия в Isis Proteus (Исходное состояние, в таблицу расписаний введены два интервала включения/выключения системы полива)

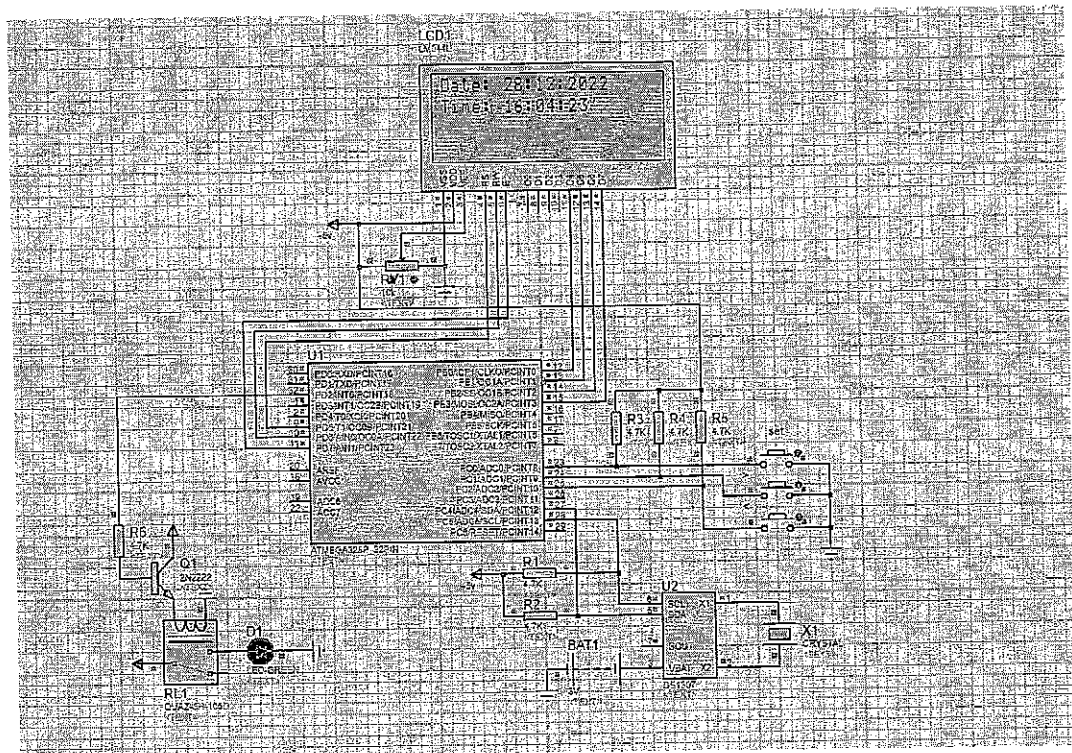


Рисунок А3 – Модель изделия в Isis Proteus (текущее время меньше времени включения полива на первом интервале, полив выключен (D1 не горит))

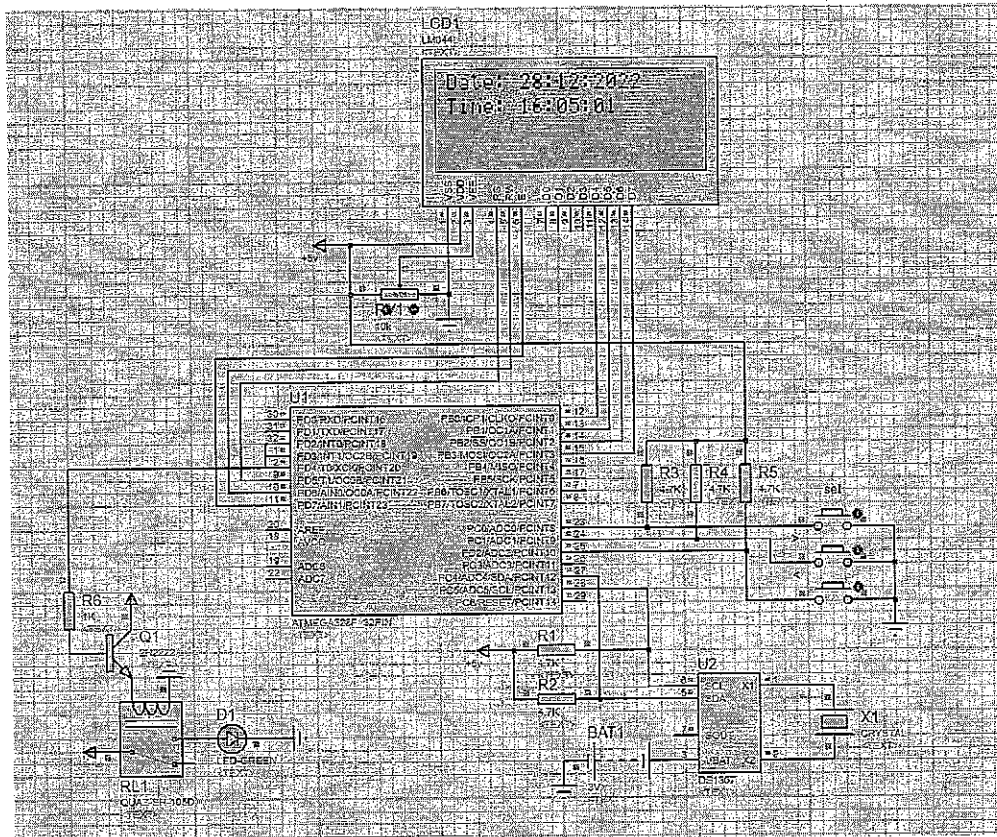


Рисунок А4 – Модель изделия в Isis Proteus (время включения полива на первом интервале превышено, полив включен (D1 горит))

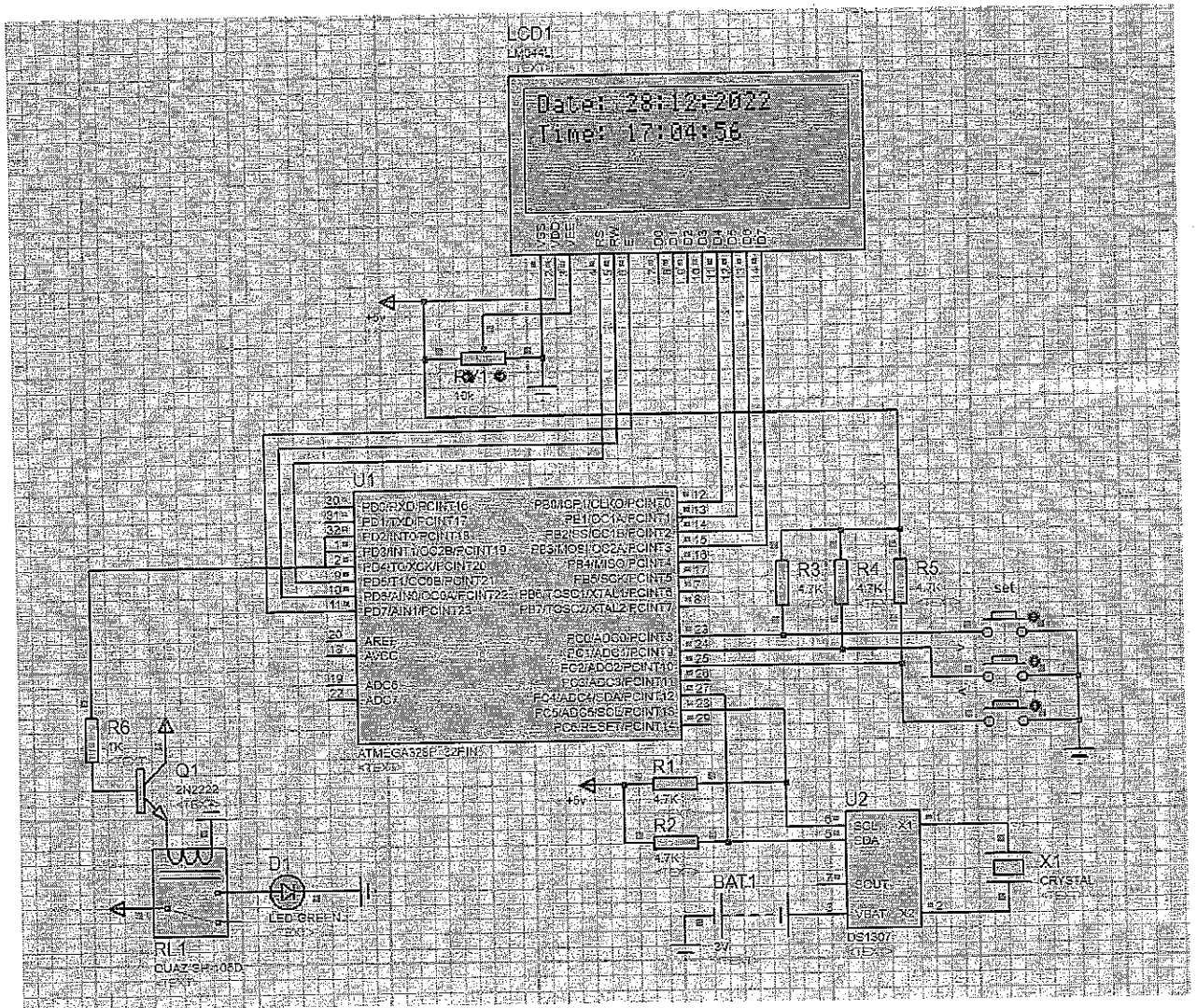


Рисунок А5 – Модель изделия в Isis Proteus (время включения полива на первом интервале истекает, полив включен (D1 горит))



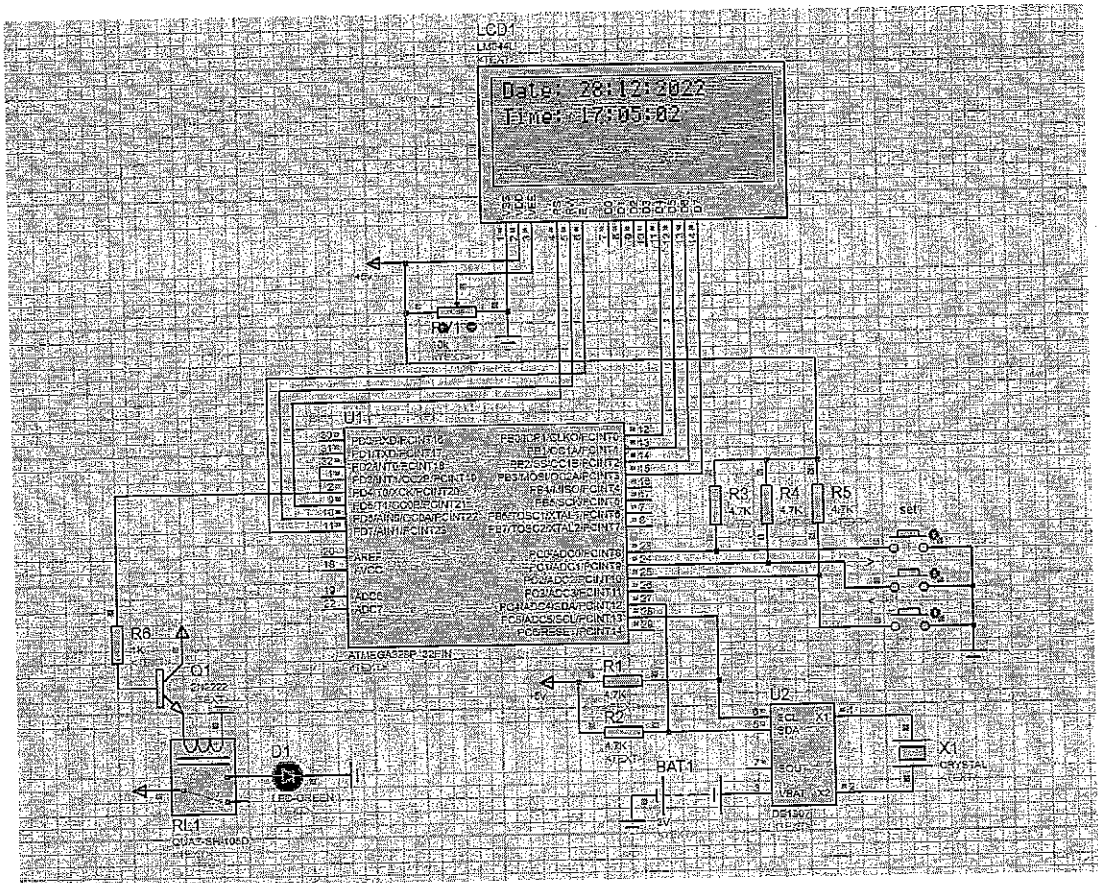


Рисунок А6 – Модель изделия в Isis Proteus (время выключения полива на первом интервале превышено, полив выключен (D1 не горит))

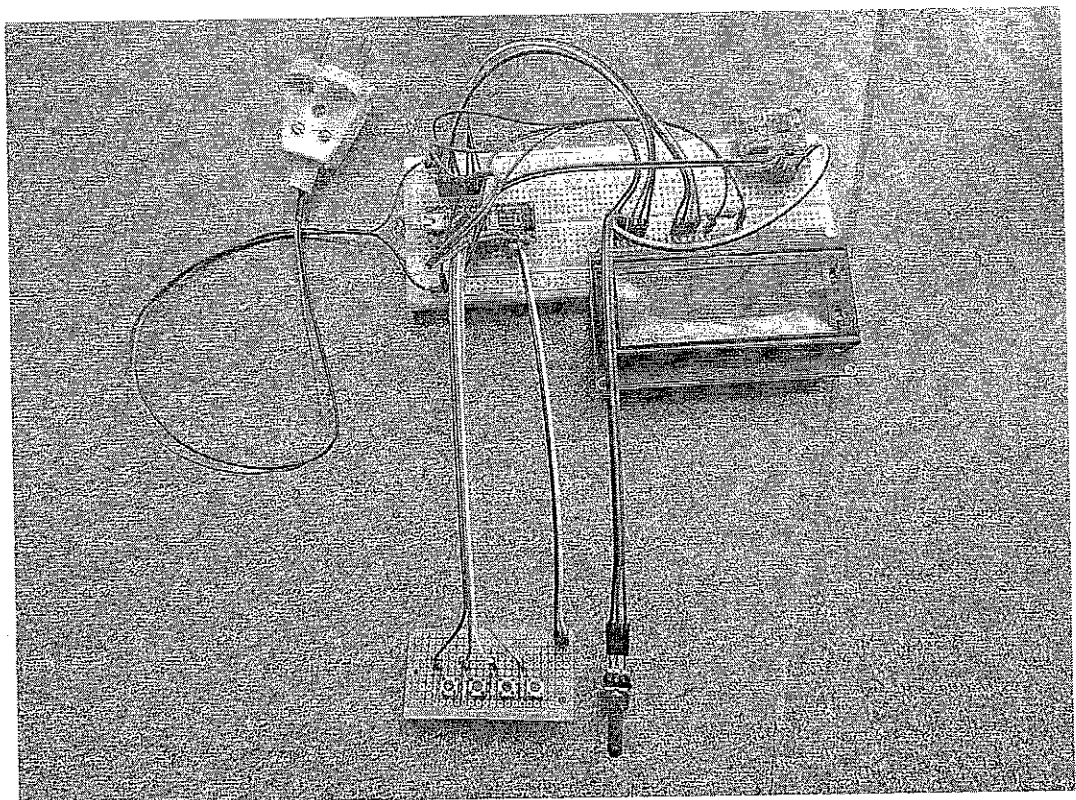


Рисунок А7 – Внешний вид макета изделия



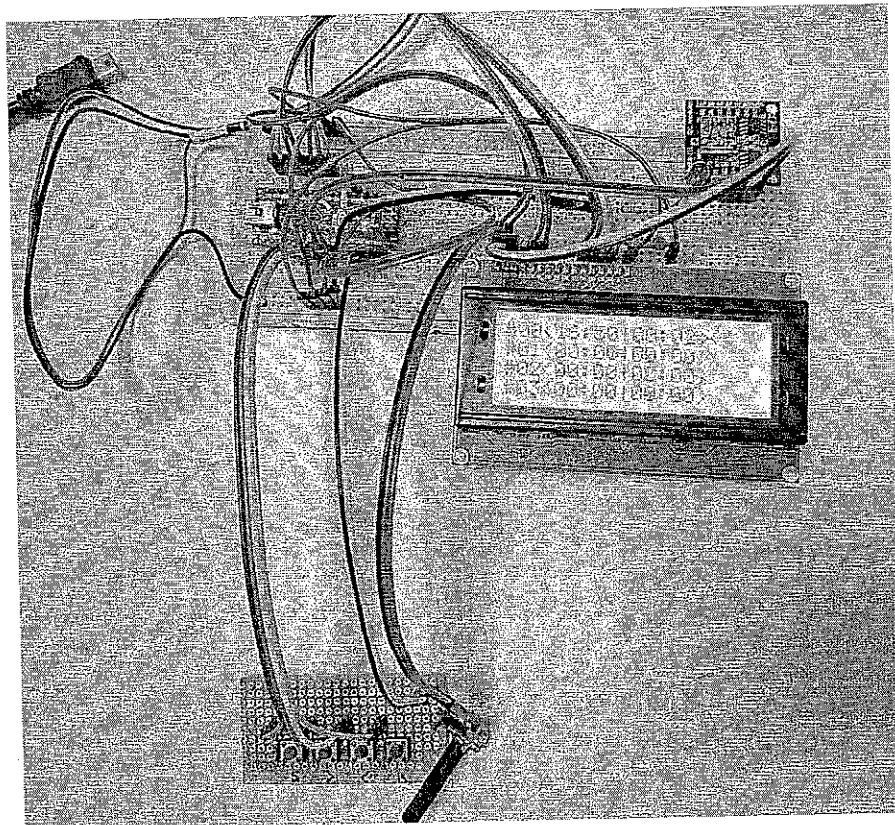


Рисунок А8 – Тестирование изделия (Исходное состояние, в таблицу расписаний введены начало интервала включения (16:50) и длительность включения системы полива (00:02))

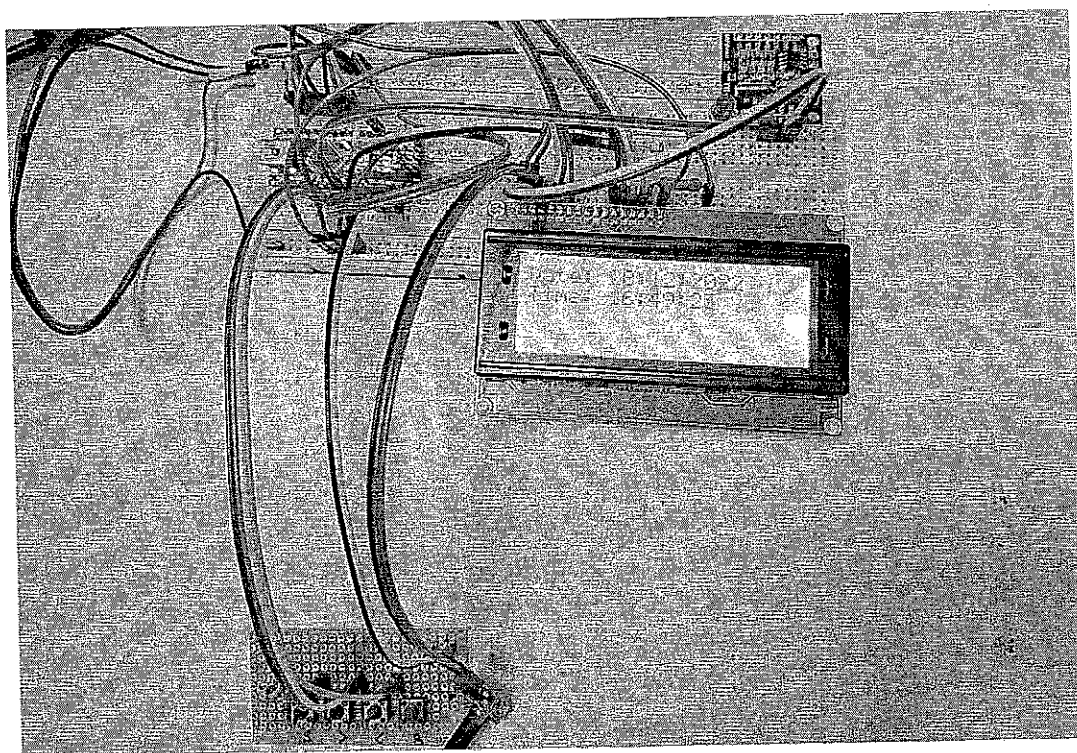


Рисунок А9 – Тестирование изделия (текущее время меньше времени включения полива на первом интервале (16:50), полив выключен (D1 не горит))

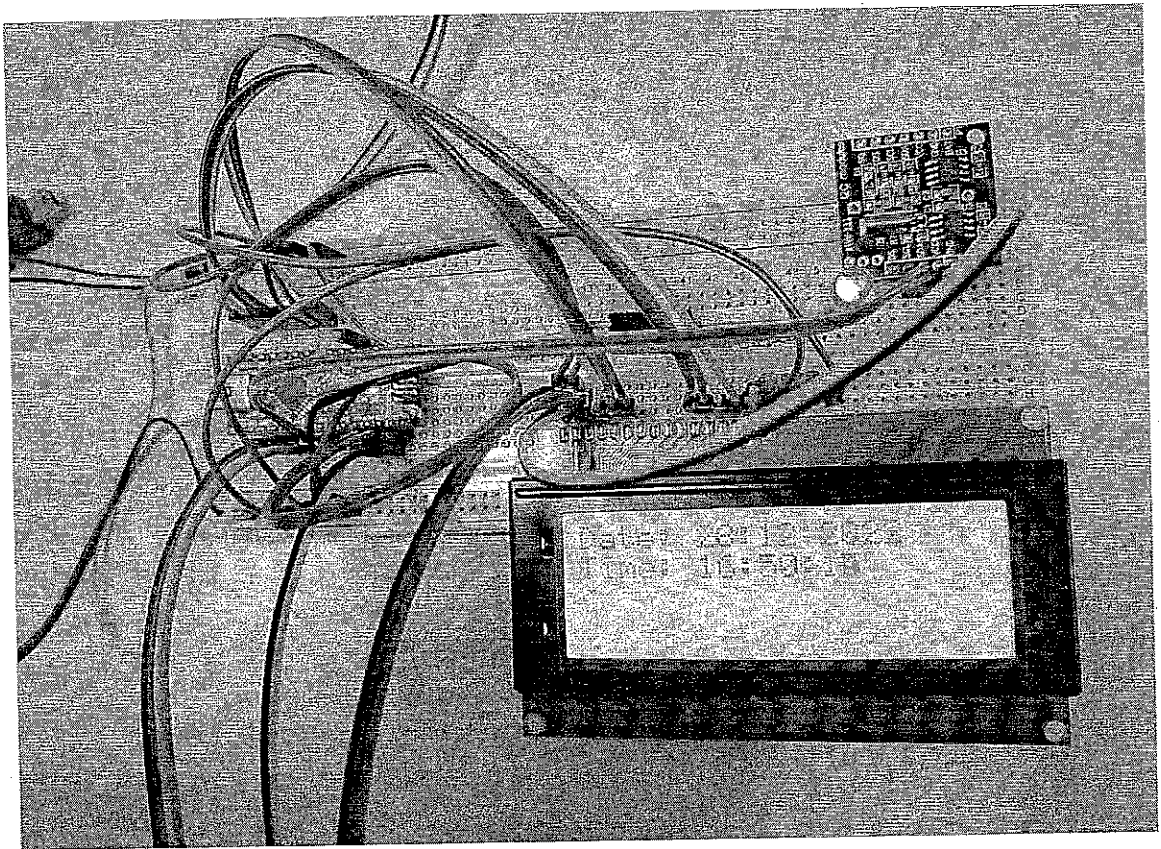


Рисунок А10 – Тестирование изделия (время включения полива на первом интервале (16:50) превышено, полив включен (D1 горит))

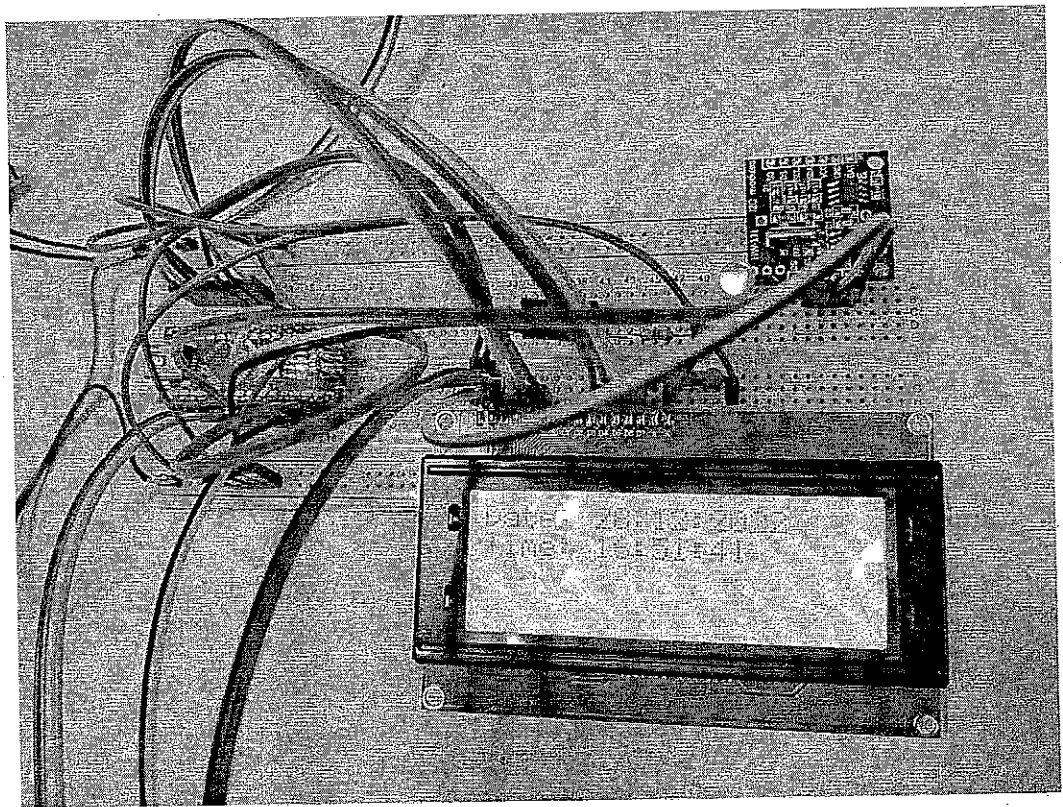


Рисунок А11 – Тестирование изделия (время выключения полива на первом интервале (16:52) истекает, полив включен (D1 горит))

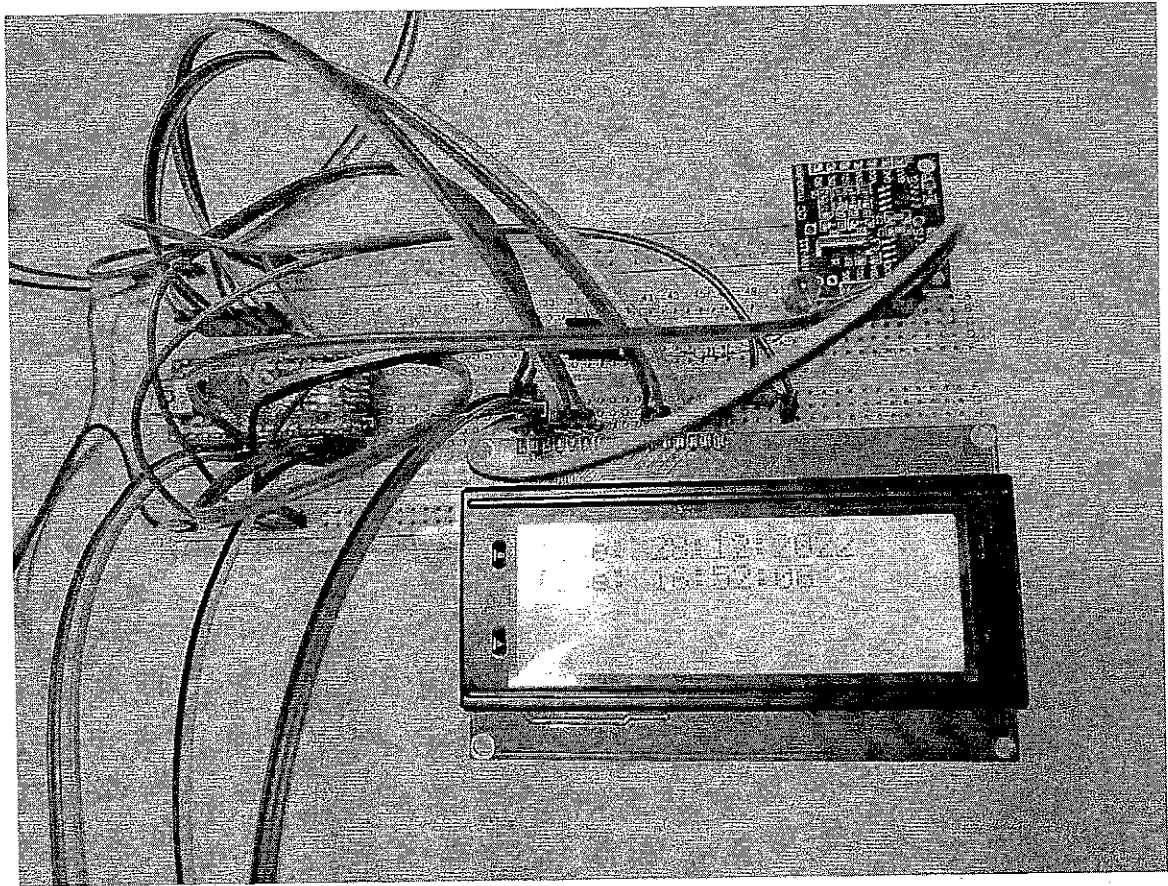
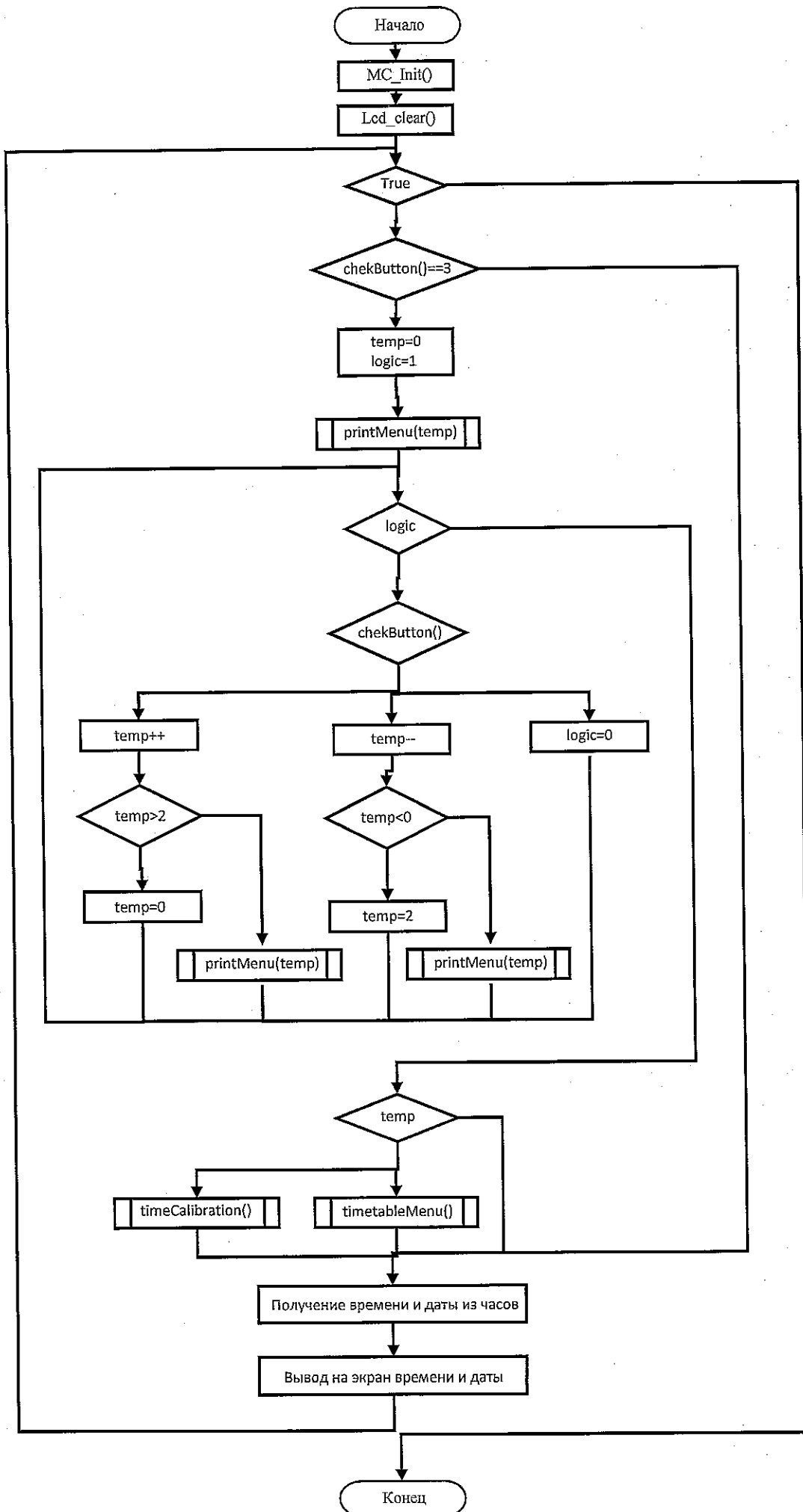


Рисунок А12 – Тестирование изделия (время выключения полива на первом интервале (16:52) превышено, полив выключен (D1 не горит))



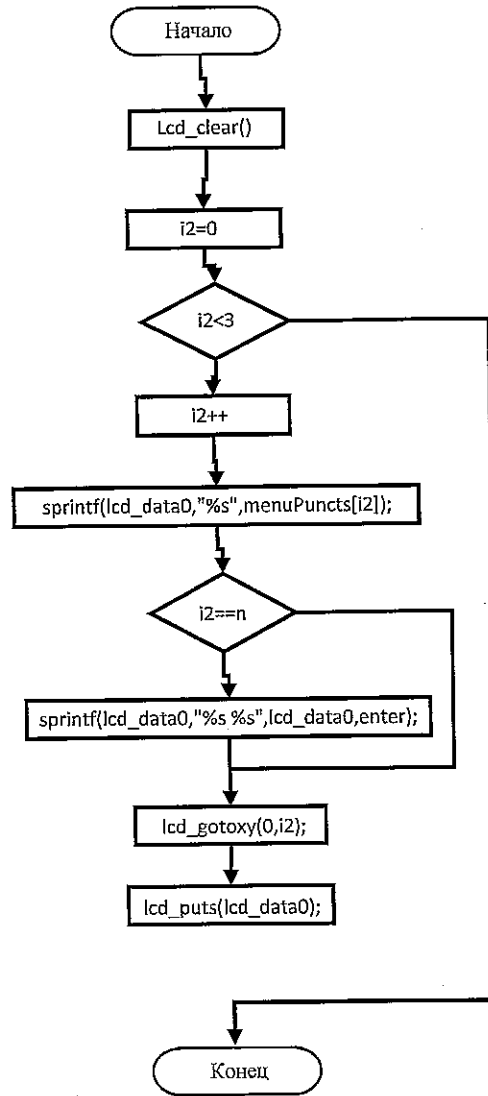


Рисунок 3 – Функция printMenu

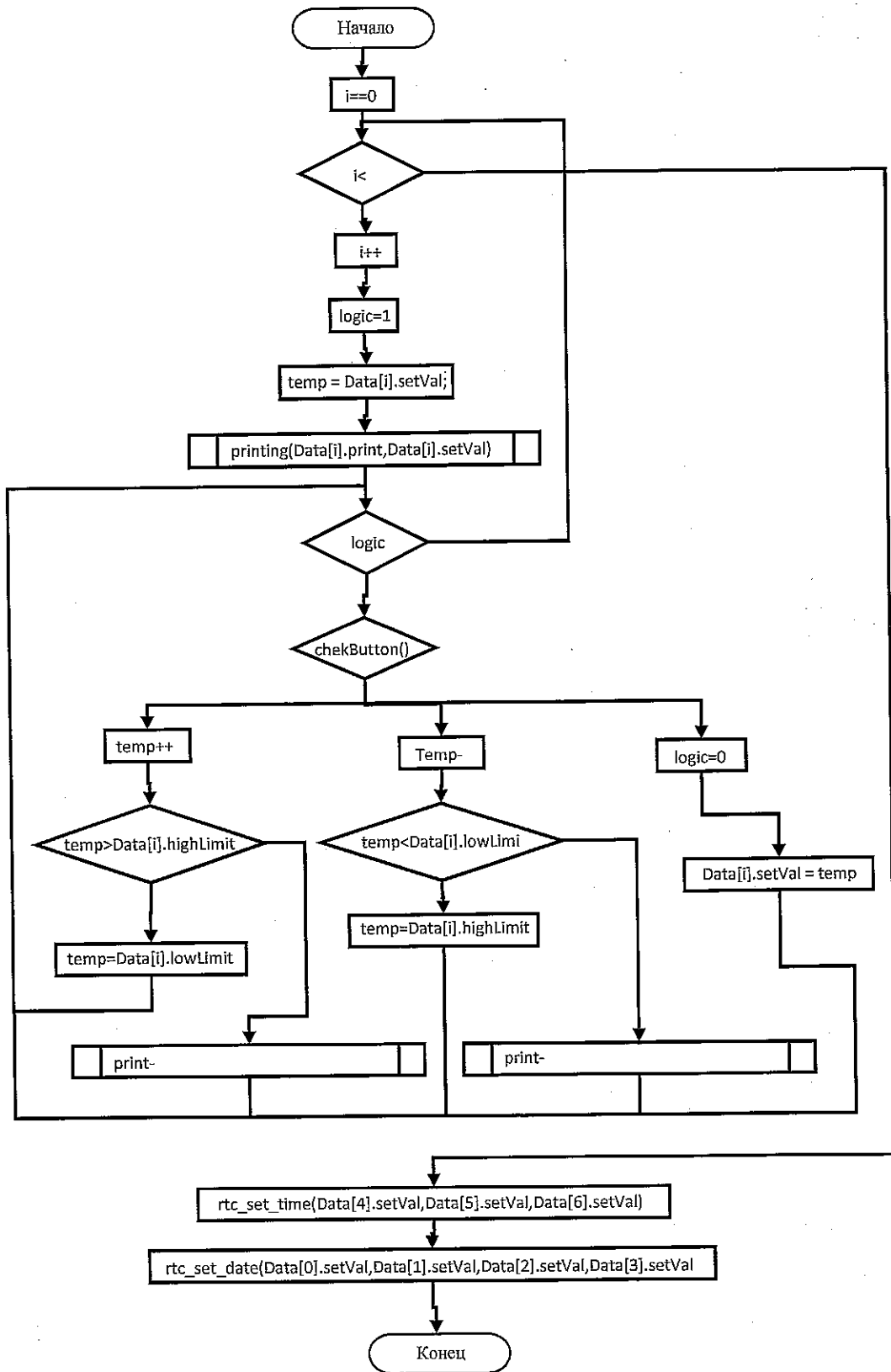


Рисунок 4 – Функция timeCalibration



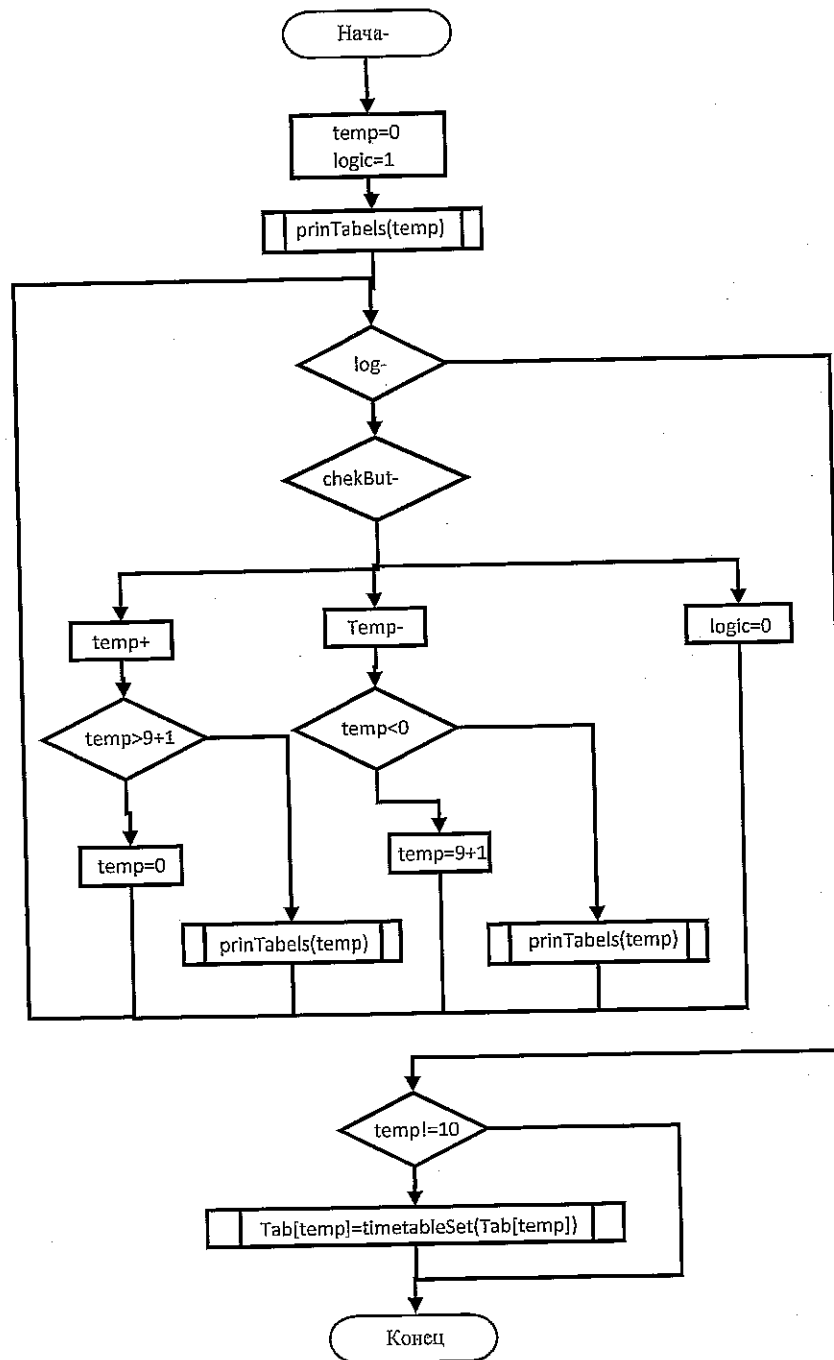


Рисунок 5 – Функция timetablesMenu

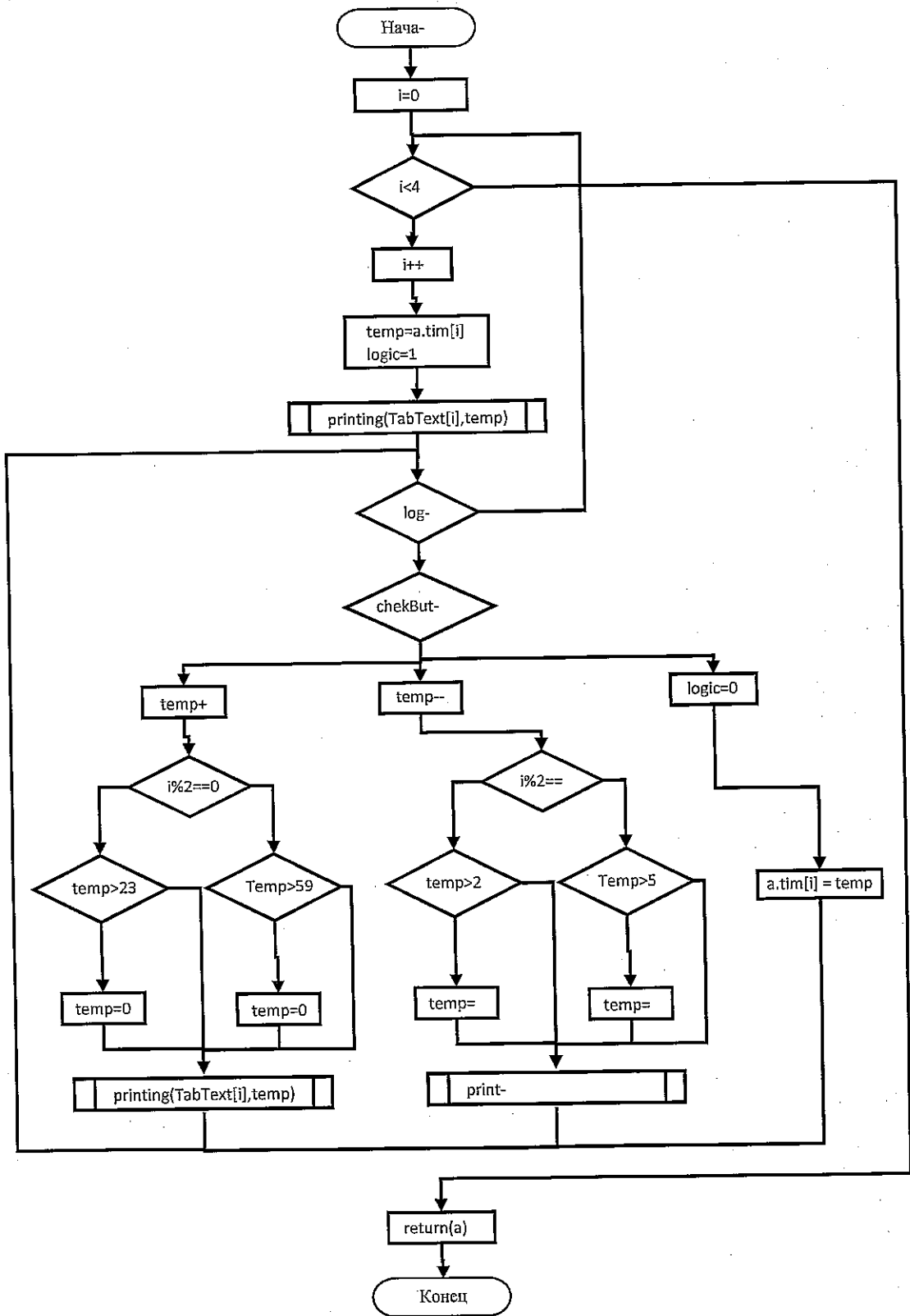


Рисунок 6 - Функция timetableSet



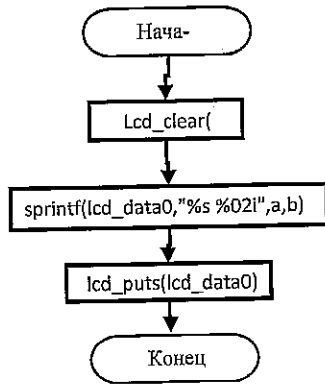


Рисунок 7 – Функция printing

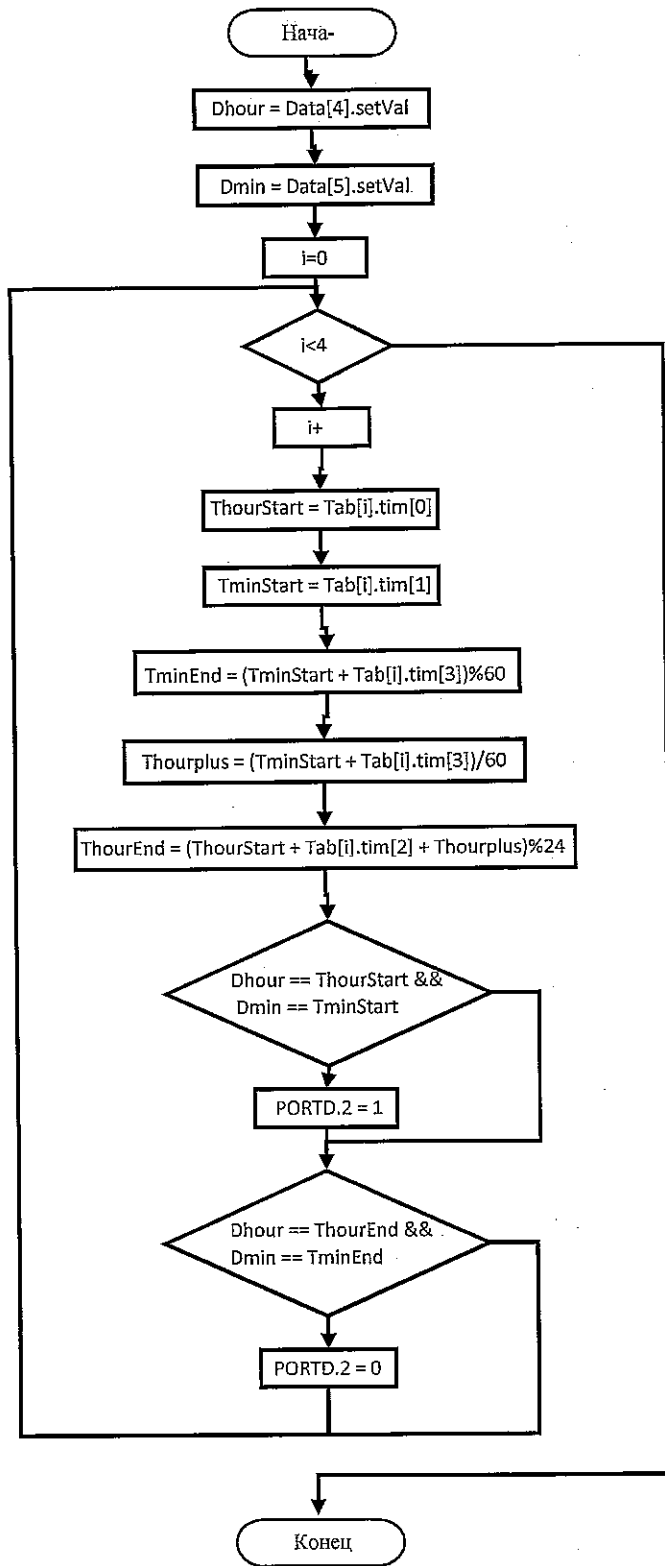


Рисунок 8 – Функция Timer

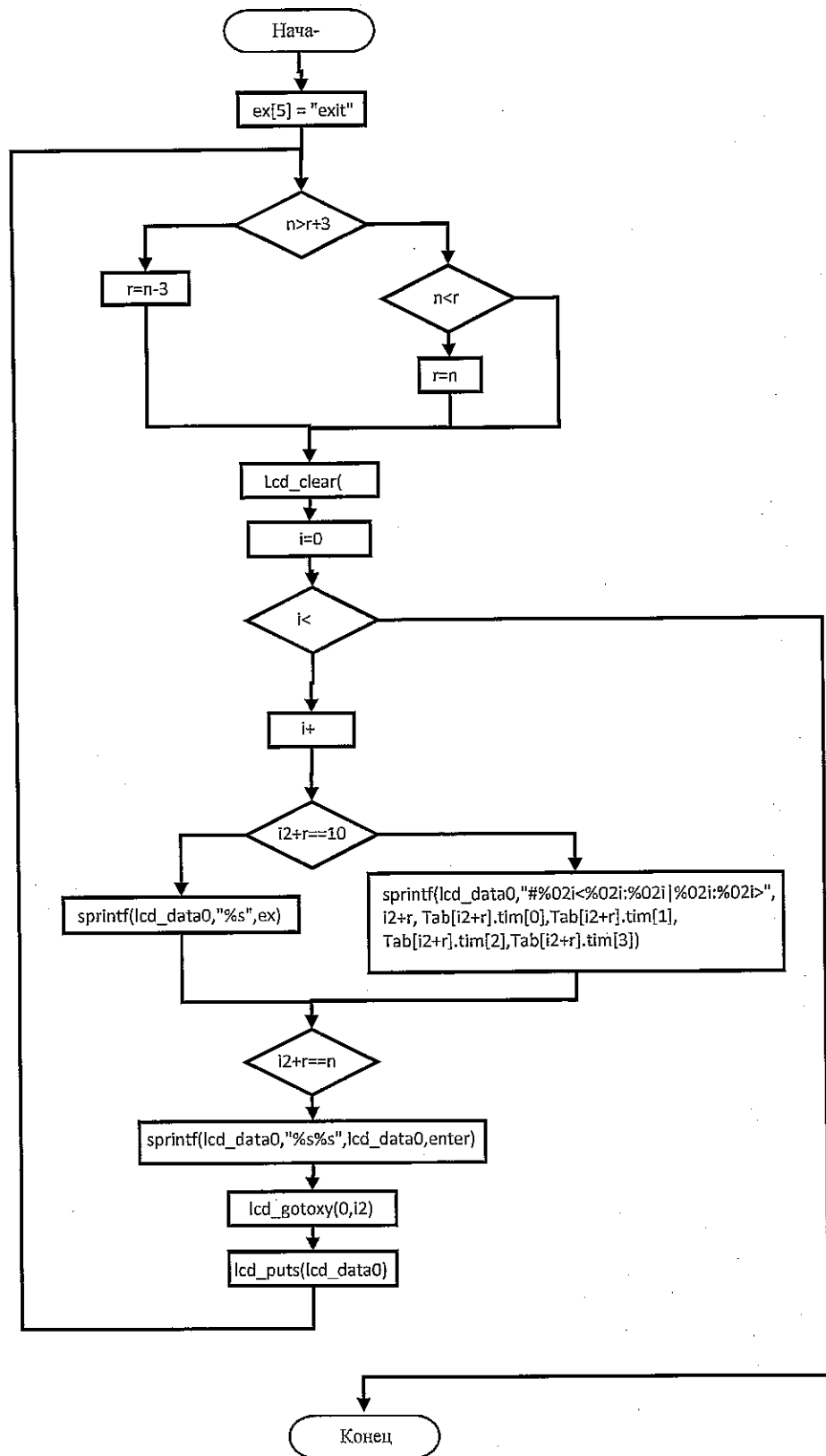


Рисунок 9 – Функция printTables

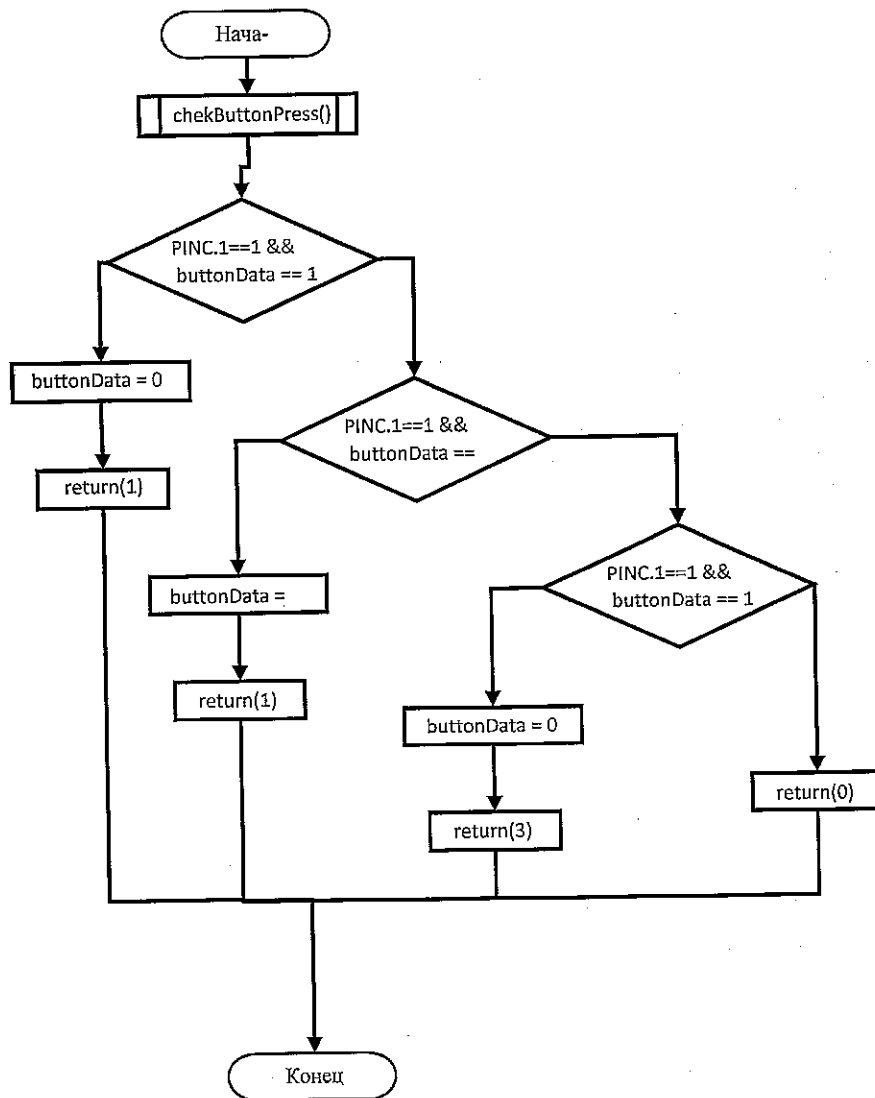


Рисунок 10 – Функция chekButton

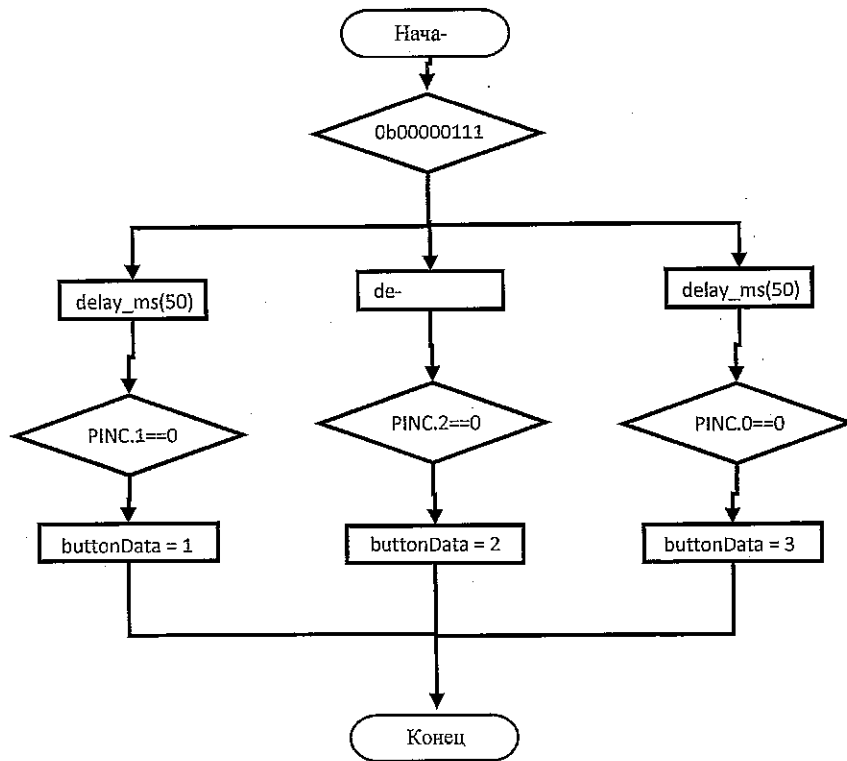


Рисунок 11 – Функция checkButtonPress

## Листинг управляющей программы

```
/******  
Chip type      : ATmega328P  
AVR Core Clock frequency: 16,000000 MHz  
*****/  
#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>  
  
// Объявление глобальных переменных  
char lcd_data0[20];  
char lcd_data1[20];  
//float dat = 21.0;  
  
int temp;  
bit logic;  
char i;  
char r=0;  
  
char menuPuncts[3][17] = {  
    "time calibration",  
    "timetables",  
    "exit"  
};  
  
char enter[3] = "<<";  
  
typedef struct  
{  
    char print[12];  
    char lowLimit;  
    char highLimit;  
    char setVal;  
} TIME;  
  
TIME Data[7] = {  
    {"Week_day:",1,7},  
    {"Day:",1,30},  
    {"Month:",1,12},  
    {"Year:",0,99},  
    {"Hour:",0,23},  
    {"Min:",0,59},  
    {"Sec:",0,59}  
};  
  
typedef struct  
{  
    char tim[4];  
}  
} TABLE;  
  
TABLE Tab[10];  
  
char TabText[4][12] = {  
    "start Hour",  
    "start min",  
    "stop Hour",  
    "stop min"  
};  
  
void printing(char a[12], char b){  
    lcd_clear(); // Очистка экрана  
    sprintf(lcd_data0,"%s %02i",a,b); //Запись в буфер  
    lcd_puts(lcd_data0); // Вывод на экран  
}  
  
void prinMenu( char n){
```

```

char i2;
lcd_clear();
for(i2=0;i2<3;i2++){
    sprintf(lcd_data0,"%s",menuPuncts[i2]);
    if(i2==n)
        sprintf(lcd_data0,"%s %s",lcd_data0,enter);
    lcd_gotoxy(0,i2);
    lcd_puts(lcd_data0);
}
}

void prinTabels(char n){
    char ex[5] = "exit";
    char i2;
    if(n>r+3)
        r=n-3;
    else if(n<r)
        r=n;
    lcd_clear();
    for(i2=0;i2<4;i2++){
        if(i2+r==10)
            sprintf(lcd_data0,"%s",ex);
        else
            sprintf(lcd_data0,"#%02i<%02i:%02i%02i:%02i>",
                i2+r,
                Tab[i2+r].tim[0],Tab[i2+r].tim[1],
                Tab[i2+r].tim[2],Tab[i2+r].tim[3]);
        if(i2+r==n)
            sprintf(lcd_data0,"%s%s",lcd_data0,enter);
        lcd_gotoxy(0,i2);
        lcd_puts(lcd_data0);
    }
}

char buttonData = 0;

void chekButtonPress(void){
    switch (PINC&0b00000111) // Отслеживаем нажатие кнопок
    {
        case 0x05: delay_ms(50); if(PINC.1==0) buttonData = 1; break;
        case 0x03: delay_ms(50); if(PINC.2==0) buttonData = 2; break;
        case 0x06: delay_ms(50); if(PINC.0==0) buttonData = 3; break;
    };
}

char chekButton(void){
    chekButtonPress();
    if(PINC.1==1 && buttonData == 1){buttonData = 0; return(1);}
    else if(PINC.2==1 && buttonData == 2) {buttonData = 0; return(2);}
    else if(PINC.0==1 && buttonData == 3) {buttonData = 0; return(3);}
    else return(0);
}

void MC_Init(void);
void timeCalibration(void);
void timetablesMenu(void);
TABLE timetableSet(TABLE a);

void main(void)
{
    MC_Init();//Инициализация периферии
    // Объявление локальных переменных

    lcd_clear();//очистка экрана
    while (1)
    {
        if(chekButton()==3){
            #asm("cli")
            temp=0;
            logic=1;
            prinMenu(temp);
        }
    }
}

```

```

while (logic){
    switch (chekButton()) // Отслеживаем нажатие кнопок
    {
        case 1: temp++; if (temp>2)temp=0; prinMenu(temp); break;
        case 2: temp--; if (temp<0)temp=2; prinMenu(temp); break;
        case 3: logic = 0; break;
    };
}
switch (temp)
{
    case 0: timeCalibration(); break;
    case 1: timetablesMenu(); break;
    case 2: break;
}
#asm("sei")
}

rtc_get_date(&Data[0].setVal,&Data[1].setVal,&Data[2].setVal,&Data[3].setVal); //считать дату
rtc_get_time(&Data[4].setVal,&Data[5].setVal,&Data[6].setVal); //считать время
sprintf(lcd_data0,"Date: %02i:%02i:%02i  ",Data[1].setVal,Data[2].setVal,Data[3].setVal+2000); // Запись в буфер
значения даты
//sprintf(lcd_data0,"Time: %u%u:%u%u:%u%u",hour/10,hour%10,min1/10,min1%10,sec/10,sec%10); // Запись в
буфер значения времени
sprintf(lcd_data1,"Time: %02i:%02i:%02i  ",Data[4].setVal,Data[5].setVal,Data[6].setVal); // Запись в буфер
значения времени
lcd_clear(); //очистка экрана
lcd_gotoxy(0,0);
lcd_puts(lcd_data0); //Выводим на экран текущую дату
lcd_gotoxy(0,1);
lcd_puts(lcd_data1); //Выводим на экран текущее время
delay_ms(200);
}
}

void timetablesMenu(void){
    temp=0;
    logic=1;
    prinTabels(temp);
    while (logic)
    {
        switch (chekButton()) // Отслеживаем нажатие кнопок
        {
            case 1: temp++; if (temp>9+1){temp=0;}; prinTabels(temp); break;
            case 2: temp--; if (temp<0){temp=9+1;}; prinTabels(temp); break;
            case 3: logic = 0; break;
        };
    }
    if(temp != 10)
        Tab[temp]=timetableSet(Tab[temp]);
}

TABLE timetableSet(TABLE a){
    for(i=0;i<4;i++){
        logic = 1;
        temp = a.tim[i];
        printing(TabText[i],temp);
        while (logic){
            switch (chekButton()) // Отслеживаем нажатие кнопок
            {
                case 1: temp++; if(i%2==0){if (temp>23){temp=0;}}else{if (temp>59){temp=0;}};
                printing(TabText[i],temp); break; //Если нажата кнопка ">" то увеличиваем колчество
                case 2: temp--; if(i%2==0){if (temp<0){temp=23;}}else{if (temp<0){temp=59;}};
                printing(TabText[i],temp); break; //Если нажата кнопка "<" то уменьшаем колчество
                case 3: logic = 0; a.tim[i] = temp; break; //Если нажата кнопка "Set", переходим дальше
            };
        }
    }
    return (a);
}

interrupt [TIM1_OVF] void timer1_ovf_isr(void){

```



```

char Dhour = Data[4].setVal;
char Dmin = Data[5].setVal;

for(i=0;i<10;i++){
    char ThourStart = Tab[i].tim[0];
    char TminStart = Tab[i].tim[1];

    char TminEnd = (TminStart + Tab[i].tim[3])%60;
    char Thourplus = (TminStart + Tab[i].tim[3])/60;
    char ThourEnd = (ThourStart + Tab[i].tim[2] + Thourplus)%24;

    if(Dhour == ThourStart && Dmin == TminStart){
        PORTD.2 = 1;
    }
    if(Dhour == ThourEnd && Dmin == TminEnd){
        PORTD.2 = 0;
    }
}

void timeCalibration(void){

    for(i=0;i<7;i++){
        logic = 1;
        temp = Data[i].setVal;
        printing(Data[i].print,Data[i].setVal);
        while (logic){
            switch (chekButton()) // Отслеживаем нажатие кнопок
            {
                case 1: temp++; if (temp>Data[i].highLimit){temp=Data[i].lowLimit}; printing(Data[i].print,temp);
                break; //Если нажата кнопка ">" то увеличиваем колчество
                case 2: temp--; if (temp<Data[i].lowLimit){temp=Data[i].highLimit}; printing(Data[i].print,temp);
                break; //Если нажата кнопка "<" то уменьшаем колчество
                case 3: logic = 0; Data[i].setVal = temp; break; //Если нажата кнопка "Set", переходим дальше
            };
        }
        rtc_set_time(Data[4].setVal,Data[5].setVal,Data[6].setVal);
        rtc_set_date(Data[0].setVal,Data[1].setVal,Data[2].setVal,Data[3].setVal);
    }

    void MC_Init(void)
    {
        // Crystal Oscillator division factor: 1
        #pragma optimize-
        CLKPR=(1<<CLKPCE);
        CLKPR=(0<<CLKPCE) | (0<<CLKPS3) | (0<<CLKPS2) | (0<<CLKPS1) | (0<<CLKPS0);
        #ifdef OPTIMIZE_SIZE_
        #pragma optimize+
        #endif

        // Input/Output Ports initialization
        // Port B initialization
        // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
        DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) | (0<<DDB2) | (0<<DDB1) | (0<<DDB0);
        // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
        PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) |
        (0<<PORTB1) | (0<<PORTB0);

        // Port C initialization
        // Function: Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
        DDRC=(0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);
        // State: Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
        PORTC=(0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) |
        (0<<PORTC0);

        // Port D initialization
        // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
        DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (1<<DDD3) | (1<<DDD2) | (0<<DDD1) |
        (0<<DDD0);
        // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
    }
}

```

```
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);
```

```
// Timer/Counter 0 initialization  
// Clock source: System Clock  
// Clock value: Timer 0 Stopped  
// Mode: Normal top=0xFF  
// OC0A output: Disconnected  
// OC0B output: Disconnected  
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;
```

```
// Timer/Counter 1 initialization  
// Clock source: System Clock  
// Clock value: Timer1 Stopped  
// Mode: Normal top=0xFFFF  
// OC1A output: Disconnected  
// OC1B output: Disconnected  
// Noise Canceler: Off  
// Input Capture on Falling Edge  
// Timer1 Overflow Interrupt: Off  
// Input Capture Interrupt: Off  
// Compare A Match Interrupt: Off  
// Compare B Match Interrupt: Off  
TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);  
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (1<<CS12) | (0<<CS11) | (0<<CS10);  
TCNT1H=0x00;  
TCNT1L=0x00;  
ICR1H=0x00;  
ICR1L=0x00;  
OCR1AH=0x00;  
OCR1AL=0x00;  
OCR1BH=0x00;  
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization  
// Clock source: System Clock  
// Clock value: Timer2 Stopped  
// Mode: Normal top=0xFF  
// OC2A output: Disconnected  
// OC2B output: Disconnected  
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;
```

```
// Timer/Counter 0 Interrupt(s) initialization  
TIMSK0=(0<<OCIE0B) | (0<<OCIE0A) | (0<<TOIE0);
```

```
// Timer/Counter 1 Interrupt(s) initialization  
TIMSK1=(0<<ICIE1) | (0<<OCIE1B) | (0<<OCIE1A) | (1<<TOIE1);
```

```
// Timer/Counter 2 Interrupt(s) initialization  
TIMSK2=(0<<OCIE2B) | (0<<OCIE2A) | (0<<TOIE2);
```

```
// External Interrupt(s) initialization  
// INT0: Off  
// INT1: Off  
// Interrupt on any change on pins PCINT0-7: Off  
// Interrupt on any change on pins PCINT8-14: Off  
// Interrupt on any change on pins PCINT16-23: Off  
EICRA=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);  
EIMSK=(0<<INT1) | (0<<INT0);  
PCICR=(0<<PCIE2) | (0<<PCIE1) | (0<<PCIE0);
```

```
// USART initialization  
// Communication Parameters: 8 Data, 1 Stop, No Parity
```

```

// USART Receiver: On
// USART Transmitter: On
// USART0 Mode: Asynchronous
// USART Baud Rate: 9600
UCSR0A=(0<<RXCO) | (0<<TXCO) | (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;

// Analog Comparator initialization
// Analog Comparator: Off
// The Analog Comparator's positive input is
// connected to the AIN0 pin
// The Analog Comparator's negative input is
// connected to the AIN1 pin
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
ADCSRB=(0<<ACME);
// Digital input buffer on AIN0: On
// Digital input buffer on AIN1: On
DIDR1=(0<<AIN0D) | (0<<AIN1D);

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

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

// TWI initialization
// TWI disabled
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

// Bit-Banged I2C Bus initialization
// I2C Port: PORTC
// I2C SDA bit: 4
// I2C SCL bit: 5
// Bit Rate: 100 kHz
// Note: I2C settings are specified in the
// Project|Configure|C Compiler|Libraries|I2C menu.
i2c_init();

// DS1307 Real Time Clock initialization
// Square wave output on pin SQW/OUT: Off
// SQW/OUT pin state: 0
rtc_init(0,0,0);

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTD Bit 5
// RD - PORTD Bit 6
// EN - PORTD Bit 7
// D4 - PORTB Bit 0
// D5 - PORTB Bit 1
// D6 - PORTB Bit 2
// D7 - PORTB Bit 3
// Characters/line: 20
lcd_init(20);

asm("sei")
}


```

Министерство науки и высшего образования Российской Федерации

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

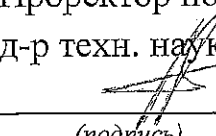
СОГЛАСОВАНО

Начальник отдела ОНиПКРС


  
(подпись) Е.М. Димитриади  
« 14 » 06 20 23 г.

УТВЕРЖДАЮ

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

  
(подпись) А.В. Космынин  
« 19 » 06 20 23 г.

Декан ФЭУ

  
(подпись) А.С. Гудим  
« 20 » 06 20 23 г.

АКТ

о приемке в эксплуатацию проекта

Аппаратно-программный комплекс

«Микроконтроллерная система капельного полива растений»

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,
- И.А. Иванова – ОМРБ-1

составила акт о нижеследующем:

«Исполнитель» передает проект «Микроконтроллерная система  
капельного полива растений», в составе:

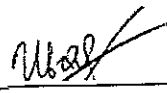
1. Блок управления системы капельного полива.
2. Рабочие программы управления изделием.
3. Паспорт изделия

Руководитель проекта

 20.12.2022  
(подпись, дата)


В.А. Егоров

Исполнитель проекта

  
(подпись, дата)

И.А. Иванова

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