## Курсовой проект

на тему:

«Разработка термометра с автоматическим контролем температуры

на базе микроконтроллера AТ90S2313 с применением термостата DS1620»

**Введение**

Измерение, контроль и регулирование температуры является одной из неотъемлемых и важных задач в современном мире. Такая задача стоит и перед промышленностью, и перед сельским хозяйством, и в быту и даже в области высоких технологий. В разных случаях задача регулирования температуры имеет свою индивидуальную цель и метод решения.

Регулирование температуры различных газообразных, сыпучих и жидких сред, в том числе воздуха и воды является задачей и для дорожно–строительной отрасли промышленности. Примером может служить регулирование температуры приготовления битумной смеси, асфальта и т.д.

Возлагать на человека задачу контроля и регулирования температуры технологических процессов в эпоху высоких компьютерных технологий просто не рационально. На помощь человеку пришли различные цифровые датчики и регуляторы температуры с использованием микропроцессорной техники.

В данном курсовом проекте рассматривается проектирование и создание цифрового регулятора температуры на базе микросхемы-термометра DS1620, который совмещает температурный датчик, схему управления и АЦП в одном кристалле, и микроконтроллера фирмы Atmel AT90S2313.

**1. Анализ задачи. Выбор компонентов**

Наша задача состоит в поддержании температуры в заданном диапазоне значений. Коридор значений температуры задается оператором с клавиатуры. На ЖКИ необходимо выводить данные о текущей температуре, о верхнем пределе и о нижнем пределе (поочередно, в зависимости от выбранного оператором режима). Температурный датчик должен сообщать о нахождении температуры в заданном коридоре свечением светодиода на одном из своих выводов (Tcom). Также цифровой термостат будет осуществлять управление исполнительными устройствами (нагревателем и охладителем) подачей сигнала высокого уровня на соответствующие выводы (Thigh и Tlow). Одновременно включенными оба исполнительных механизма быть не должны. Задачи по управлению термостатом, по выводу информации на ЖКИ, по обработке информации с термостата и с клавиатуры оператора возлагаются на микроконтроллер.

**1.1 Микроконтроллер**

AT90S2313 - экономичный 8 битовый КМОП микроконтроллер, построенный с использованием расширенной RISC архитектуры AVR. Исполняя по одной команде за период тактовой частоты, AT90S2313 имеет производительность около 1MIPS на МГц, что позволяет разработчикам создавать системы оптимальные по скорости и потребляемой мощности.

В основе ядра AVR лежит расширенная RISC архитектура, объединяющая развитый набор команд и 32 регистра общего назначения. Все 32 регистра непосредственно подключены к арифметико-логическому устройству (АЛУ), что дает доступ к любым двум регистрам за один машинный цикл.

Подобная архитектура обеспечивает десятикратный выигрыш в эффективности кода по сравнению с традиционными CISC микроконтроллерами.

AT90S2313 предлагает следующие возможности: 2кБ загружаемой флэш памяти; 128 байт EEPROM; 15 линий ввода/вывода общего назначения; 32 рабочих регистра; настраиваемые таймеры/счетчики с режимом совпадения; внешние и внутренние прерывания; программируемый универсальный последовательный порт; программируемый сторожевой таймер со встроенным генератором; SPI последовательный порт для загрузки программ; два выбираемых программно режима низкого энергопотребления. Холостой режим (Idle Mode) отключает ЦПУ, оставляя в рабочем состоянии регистры, таймеры/счетчики, SPI порт и систему прерываний. Экономичный режим (Power Down Mode) сохраняет содержимое регистров, но отключает генератор, запрещая функционирование всех встроенных устройств до внешнего прерывания или аппаратного сброса.

Микросхемы производятся с использованием технологии энергонезависимой памяти высокой плотности фирмы Atmel. Загружаемая флэш память на кристалле может быть перепрограммирована прямо в системе через последовательный интерфейс SPI или доступным программатором энергонезависимой памяти. Объединяя на одном кристалле усовершенствованный 8-битовый RISC процессор с загружаемой флэш–памятью, AT90S2313 является мощным микроконтроллером, который позволяет создавать достаточно гибкие и эффективные по стоимости устройства. AT90S2313 поддерживается полной системой разработки, включающей в себя макроассемблер, программный отладчик/симулятор, внутрисхемный эмулятор и отладочный комплект.

**1.2 Цифровой датчик-термометр DS1620**

Для измерения температуры в цифровых устройствах необходимы температурный датчик, схема управления и АЦП. Последние модели температурных датчиков совмещают эти компоненты в одном кристалле.

Микросхема DS1620 (Dallas RS218-3810) – это девятиразрядный термометр и термостат, служащий для измерения и отображения температуры (рис. 1). Он имеет три выхода, которые используются при работе микросхемы в режиме термостата. Настройки выходного сигнала можно запрограммировать и сохранить во внутренней энергонезависимой памяти. Устройство измеряет температуру от –55 до +12˚С шагом 0,5˚С, преобразование занимает 1 с.

Регистр состояния и управляющая логика

Температурный датчик

Триггер максимальной температуры

Триггер минимальной температуры

Выходной компаратор

Обмен данными и сброс

7

6

5

1

2

3

Tlow

Thigh

Tcom

DQ

CLK

GND

Vdd

Thigh

Tlow

GND

RST

CLK

DQ

4

3

2

1

8

7

6

5

Tcom

Передача данных от микросхемы к внешнему устройству осуществляется по трехпроводной последовательной шине: СLK/СОNV (контакт 2), DQ (контакт 1) и RЕSЕТ (контакт 3). Эти выходы совместимы с уровнями ТТЛ. Тhigt (контакт 7) – выход триггера высокой температуры. Если температура превышает установленный верхний порог, то выход Тhigt сигнализирует об этом высоким уровнем и остается в таком состоянии до тех пор, пока температура не упадет ниже заданного порога. Тlow (контакт 6) – выход триггера низкой температуры. Если температура опускается ниже определенного нижнего предела, то на нем появляется сигнал высокого уровня, сохраняющийся до тех пор, пока температура не поднимется выше указанного предела. Тсоm (контакт 5) – это выход комбинированного триггера высокой и низкой температуры. Тсоm=1, когда температура превышает верхний предел, Тсоm=0, когда она опускается ниже нижнего предела. Контакты 4 и 8 соединены с отрицательным и положительным проводами источника питания. Потребляемый ток в режиме ожидания равен 1 мкА, в рабочем режиме - 1 мА.

Управление устройством осуществляется в два этапа: сначала команды управления последовательно загружаются в микросхему, а затем девятиразрядное число, соответствующее температуре, либо считывается, либо записывается. Микросхема имеет девять команд:

1. Read temp (AАh): чтение значения регистра, содержащего результат последнего измерения.

2. Start conversion T (EЕh): запуск процесса измерения температуры. Данные не передаются.

3. Stop convert T (22h): остановка измерения. Данные не передаются.

4. Write TH (01h): запись верхнего предела в триггер высокой температуры – 9 бит данных.

5. Write TL (01h): запись нижнего предела в триггер высокой температуры – 9 бит данных.

6. Read TH (A1h): чтение содержимого триггера высокой температуры – 9 бит данных.

7. Read TL (A2h): чтение содержимого триггера низкой температуры – 9 бит данных.

8. Write configuratioin (0Ch): запись настроечных данных в регистр настройки – 8 бит данных.

9. Read configuratioin (ACh): чтение настроечных данных из регистра настройки – 8 бит данных.

Настроечное слово управляет режимами работы микросхемы DS1620. Оно сохраняется в регистре настройки. Функции битов регистра пиведены ниже:

DONE THF TLF XXX CPU 1SHOT

|  |  |
| --- | --- |
| X  | любое |
| DONE | 0 – идет преобразование |
|  | 1 – преобразование завершено |
| THF | флаг высокой температуры. Если температура равна или выше верхнего предела, то бит ТНF=1. Он остается в единичном состоянии до тех пор, пока его не сбросят, записав ноль, или не отключат питание устройства |
| TLF | флаг низкой температуры. Если температура равна или ниже нижнего/предела, то бит ТLF=1. Он остается в единичном состоянии до тех пор, пока его не сбросят, записав ноль, или не отключат питание устройства |
| CPU | если СРU=0, то вход СLK/СОNV управляет началом цикла измерения; в противном случае микросхема работает в режиме обмена информацией с внешним устройством |
| 1SHOT | если 1SНОТ=1, микросхема производит один цикл измерения после поступления команды; в противном случае ИС настроена н непрерывное измерение температуры |

Данные о температуре имеют девятибитовый формат.

Дискретность представления температуры равна 1/2˚С. Некоторые соотношения между значениями температуры и выходными данными приведены ниже:

|  |  |
| --- | --- |
| +125˚С | 0 11111010 (00FA) |
| +25˚С | 0 00110010 (0032) |
| +1/2˚C | 0 00000001 (0001) |
|  0˚C | 0 00000000 (0000) |
| –1/2˚C | 1 11111111 (00FF) |
| –25˚С | 1 11001110 (01CE) |
| –55˚С | 1 10010010 (0192) |

Временные диаграммы передачи данных представлены на рис 2. Передача начинается при поступлении положительного фронта на вход RST (контакт 3). Если на этот вход подать 0, то передача прекращается. Процессами чтения и записи управляет тактирующий вход микросхемы. Один тактовый цикл состоит из отрицательного фронта и следующего за ним положительного. При записи информации состояние битов данных должно оставаться неизменным во время прохождения положительного фронта. При считывании данные выводятся из устройства по каждому отрицательному фронту тактовых импульсов. Когда на тактовом входе высокий уровень, выход DQ (контакт 1) имеет высокое сопротивление. При чтении данных младший бит передается первым. Через этот контакт можно как принимать, так и передавать данные. Схема с использованием термометра DS1620, подключенного к экспериментальной плате параллельного порта, приведена на рис. 3.

Поскольку вывод DQ, может использоваться как для чтения, так и для записи, а на экспериментальной плате параллельного порта таких двунаправленных линий нет, необходимо применять транзистор. База транзистора соединена с тактом D1. Когда ИС настроена на прием информации, данные поступают из ком пьютера на контакт D1, а затем через транзистор в инверсном виде – на микросхему. Когда она передает информацию, транзистор должен быть закрыт (это достигается подачей низкого уровня на его базу через тот же контакт), и данные поступают на контакт S1. Входы СLК/СОNV и RSТ соединены с контактами С1 и С2. После прохождения положительного фронта по входу RST микросхема настраивается на прием управляющей информации. Необходимые данные считываются с линии С1 под управлением тактовых импульсов. Если DS1620 настраивается на вывод данных, то после загрузки в нее управляющей команды D1 переходит в нулевое состояние, а биты данных последовательно выводятся под управлением тактовых импульсов и поступают на контакт S1. Если микросхема должна принимать данные, они также загружаются под управлением тактовых импульсов.

1

0

2

34

44

54

64

74

1

2

8

9

0

Загрузка команды в DS1620

Загрузка данных в DS1620

Нулевой

уровень

СLK

DQ

Команда

9-ти разрядные данные о температуре

мл. бит

ст. бит

RST

а)

1

0

2

34

44

54

64

74

1

2

8

9

0

Загрузка команды в DS1620

Загрузка данных в DS1620

СLK

DQ

Команда

9-ти разрядные данные о температуре

мл. бит

ст. бит

RST

б)

 Временные диаграммы работы микросхемы-термометра DS1620:

а – последовательность выгрузки данных; б – последовательность загрузки данных.

**1.3 Жидкокристаллический индикатор mt-10t7-7**

Данный индикатор может отображать 10 знакомест с точкой. Любой сегмент любого знакоместа можно включать и выключать независимо от остальных сегментов.



Внешний вид ЖКИ

 Структурная схема модуля.

Регистры данных в БИС делятся на две тетрады: SGx(L) и SGx(H). Запись данных в знакоместо производится за два такта: сначала в младшую тетраду, затем в старшую. Младшая тетрада отвечает за сегменты g, e, d, a, а старшая – за сегменты h, b, c, d. (см. рис.).



Запись Н вызывает высвечивание соответствующего сегмента, а запись L- его гашение.

Контрастность индикатора зависит от напряжения питания модуля. Управление контрастностью производится подключением внешнего резистора на вывод V0.

Rвнеш.=0 – МАХ контрастность.

Rвнеш.=∞(нет резистора) – MIN контрастность.

**Описание интерфейса ЖКИ модуля**

Сначала на шине выставляется адрес необходимого знакоместа, который фиксируется в регистре адреса при низком уровне на входе WRx (см.таблицу). Входы WR1 и ^WR2 защелкивают информацию, стоящую на шине, во внутренних регистрах статического типа. Внутри БИС эти входы объединены по схеме WR1 & ^WR2. Таким образом, информация запишется только при WR1 = “H” и ^WR2 = “L” одновременно. Такое решение осуществить функцию CS (выбор кристалла) при большом количестве модулей на шине, или если на шине имеются другие устройства.

При записи адреса знакоместа указатель тетрады сбрасывается в положение SGx(L). Запись данных производится в младшую тетраду при высоком уровне сигнала на входе А0 сигналом WRx. По этому же сигналу указатель тетрады данных переключается в положение SGx(H), сохраняя при этом тот же адрес знакоместа. Данные в старшую тетраду SGx(H) записываются аналогично младшей тетраде SGx(L). После записи второй тетрады содержимое регистра адрема инкрементируется и можно записывать данные в следующее знакоместо без записи адреса.

По адресу 0Fh расположен триггер блокировки шины. Запись в него DB0=”L” вызывает блокировку записи в БИС адресов и данных на 30 сигналов WRx. Разблокировка шины производится записью DB=”H” по адресу 0Fh.

После подачи питания содержимое регистров SGx не определено, поэтому при включении питания необходимо делать программную очистку регистров. Состояние триггера блокировки тоже не определено, поэтому перед началом вывода информации на индикатор необходимо произвести разблокировку шины. Разблокировка шины производится записью DB0=”H” по адресу 0Fh.

**2. Схемы сопряжения (функциональная и принципиальная)**

**2.1 Структурная схема**

Микроконтроллер

Клавиатура

Нагреватель

Датчик-термостат

Охладитель

R

ЖКИ

**2.2 Принципиальная схема**



**3. Алгоритм решения задачи**

Count =10

PortD (D2-D4)=0

M

ОСНОВНАЯ ПРОГРАММА

PortD (D2-D4)=0

+

–

+

Count –1

Count –1 = 0

+

–

–

MODE нажата

+

+

–

UP нажата

U

D

DOWN нажата

Считать температуру и отобразить

M

Начало

Настройка портов

Настройка начальных переменных

Настройка термостата

Настройка индикатора

ОСНОВНАЯ ПРОГРАММА

–

M

С=2

+

С=0

С=С+1

С

U

С=0

+

–

С=1

+

–

С=2

–

С

+

Тd=Tdmax

Тu=Tumax

С

+

+

С

Тu=Tu+1

–

–

Тu=Tu+1

Вывести на индикатор

Загрузить в термостат

D

С=0

+

–

С=1

+

–

С=2

–

С

+

Тd=Tdmin

Тu=Tumin

С

+

+

С

Тu=Tu-1

–

–

Тu=Tu-1

Вывести на индикатор

Загрузить в термостат

С

**3. Разработка программы**

Разработка программы производится на языке Assembler c параллельной отладкой средствами AVR Studio.

**Вывод**

В ходе данной курсовой работы был разработан цифровой регулятор температуры на базе термостата DS1620 и микроконтроллера AT90S2313 с рабочим диапазоном температур -55˚С до 125˚С. Поддерживается заданная температура с помощью двух исполнительных органов: «нагревателя» и «охладителя». Текущая температура, а также значение верхнего и нижнего значения температурного коридора отображается на ЖКИ в градусах по Цельсию с точностью 0,5˚С.

**Список литературы.**

1. Научно- технический журнал «Схемотехника» №2, 2001–2002 гг.

2. Пособие «Средства сопряжения с ПК»

3.Internet:

http://www.atmel.ru

http://www.telesys.users.ru

http://www.kulakov.ru

http://www.platan.ru

http://www.sensorsmag.com

http://www.ferrite.ru

**Приложение:** **программа на языке Assembler**

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\* Термостат на базе микроконтроллера AT90S2313 \*

;\* с использованием микросхемы термодатчика DS1620 \*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\* Название: Цифровой регулятор температуры

;\* Дата разработки: 20.06.02

;\* Объект: AVR микроконтроллер AT90S2313 и термостат DS1620

;\* Автор: Бендюк Максим Сергеевич

;\*

;\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.include "2313DEF.INC"

;\*\*\*\*\* Константы

.equ preset=192 ;T/C0 Preset constant (256-64)

.def result=r16 ;Result and intermediate data

.def temp=r16 ;временный регистр

.def temp1=r17 ;временный регистр

.def temp2=r18 ;временный регистр

.def temp3=r19 ;временный регистр

.def ramsize=r19 ;размер блока

.def tmode=r22 ;Регистр режима

.def key=r20 ;Регистр нажатой клавиши

.def keyc=r21 ;Счетчик для прграммы обработки нажатия клавиши

.def TMIN=r22 ;ячейка с содержимым минимальной температуры термостата

.def TMINL=r22 ;младший

.def TMINH=r23 ;старший

.def TMAX=r24 ;ячейка с содержимым максимальной температуры термостата

.def TMAXL=r24 ;младший

.def TMAXH=r25 ;старший

.def TNORM=r26 ;ячейка с содержимым текущей температуры термостата

.def TNORML=r26 ;младший

.def TNORMH=r27 ;старший

.equ DLCD=$66 ;стартовый адрес SRAM для буфера индикатора

;маладшая ячейка соответствует левому символу

;на дисплее

.def fbinL =r16;двоичное значение, младший байт

.def fbinH =r17;двоичное значение, старший байт

;биты портов

.equ K1 =3 ;бит первой кнопки

.equ K2 =4 ;бит второй кнопки

.equ K3 =5 ;бит третьей кнопки

.equ WR1 =5 ;бит WR1

.equ WR2 =6 ;бит WR2

.equ A0 =7 ;бит A0

.equ DQ =0 ;бит DQ канал данных термостата

.equ CLK =1 ;бит CLK канал синхронизации данных

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*

;\* PROGRAM START - программа начинает выполняться с этого места

;\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.cseg

.org $0000

rjmp RESET;Reset handle

.org $0020

RETI

RETI

RETI

RETI

RETI

RETI

RETI

RETI

RETI

RETI

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; настройка констант

SETCONST:

ldi temp,20

mov TMINL,temp

ldi temp,0

mov TMINH,temp

ldi temp,25

mov TMAXL,temp

ldi temp,0

mov TMAXH,temp

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; передача констант

OUTCONST:

mov temp,TMINl

mov temp1,TMINH

ldi temp2,2 ;запись нижней температуры

rcall TERMOUT2

mov temp,TMAXL

mov temp1,TMAXH

ldi temp2,1 ;запись нижней температуры

rcall TERMOUT2

ldi temp1,0xee ; инициализация термостата

rcall TERMOUT1

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Натройка портов ввода вывода

port\_set:

ldi result,$ff ;PORTB настраивается на вывод

out DDRB,result ;для работы с индикатором

ldi result,$00 ;все биты устанавливаются в ноль

out PORTB,result;PORTD настраивается для работы

ldi result,0b01000011 ; с клавиатурой и

out DDRD,result ;микросхемой термометра

ldi result,0b11111001 ; с клавиатурой и

out PORTD,result ;микросхемой термометра

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\* очистка буфера ЖКИ

CLRLCD:

ldi ramsize,10 ;очистка буфера индикатора

ldi temp,0

ldi ZH,high(DLCD)

ldi ZL,low(DLCD)

clrlcdloop:

st Z+,temp

dec ramsize

brne clrlcdloop ;если не закончили повторить

rjmp LCDOUT

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Вывод содержимого буфера на индикатор

LCDOUT:

ldi temp,low(DLCD) ;установить Z как указатель на буфер

mov ZL,temp ;индикатора

ldi temp,high(DLCD)

mov ZH,temp

ldi ramsize,10

LCDOUT1:

mov temp,ramsize ;запись в индикатор адреса символа

andi temp,0b00001111

out PORTB,temp

cbi PORTB,A0

sbi PORTB,WR1

cbi PORTB,WR1

ld temp1,Z+ ; получить байт из DLCD

mov temp,temp1

andi temp,0b00001111 ;получить младшую тетраду

sbrs temp,A0

out PORTB,temp

sbi PORTB,WR1

cbi PORTB,WR1

mov temp,temp1

swap temp

andi temp,0b00001111 ;получить старшую тетраду

sbrs temp,A0

out PORTB,temp

sbi PORTB,WR1

cbi PORTB,WR1

dec ramsize ;

brne LCDOUT1 ;если не закончили повторить

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\* вывод готовой строки в буфер и вывод буфера на ЖКИ

;\* temp - номер строки

;\* ramsize - количество выводимых символов

STRLCDD:

ldi YH,high(tcstr)

ldi YL,low(tcstr)

add temp,temp

add temp,temp

add YL,temp

ldi ramsize,4

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\* вывод готовой строки в буфер и вывод буфера на ЖКИ

;\* Y - указатель на строку temp - положение

;\* ramsize - количество выводимых символов

STRLCD:

ldi ZH,high(DLCD)

ldi ZL,low(DLCD)

strtolcdloop:

ld temp,Y+

st Z+,temp

dec ramsize

brne strtolcdloop ;если не закончили повторить

rjmp LCDOUT

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\* получение данных из термостата

;\* temp2 - команда temp - первые 8 бит temp1 - 9-ый бит

TERMIN:

ldi ramsize,8

terminloop:

sbi PIND,DQ

sbrc temp2,0

cbi PIND,DQ

sbi PIND,CLK

cbi PIND,CLK

ror temp2

dec ramsize

brne terminloop ;если не закончили повторить

ldi ramsize,0x8

sbi DDRD,DQ

termindloop:

sbi PIND,DQ

sbrc temp,0

cbi PIND,DQ

sbi PIND,CLK

cbi PIND,CLK

ror temp2

dec ramsize

brne termindloop ;если не закончили повторить

sbi PIND,DQ

sbrc temp1,0

cbi PIND,DQ

sbi PIND,CLK

cbi PIND,CLK

cbi DDRD,DQ

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\* передача одного байта данных на термостат

;\* temp2 - команда

TERMOUT1:

ldi ramsize,8

termout1loop:

sbi PIND,DQ

sbrc temp2,0

cbi PIND,DQ

sbi PIND,CLK

cbi PIND,CLK

ror temp2

dec ramsize

brne termout1loop ;если не закончили повторить

ldi ramsize,0x8

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\* передача одного байта данных на термостат и 9 бит

;\* temp2 - команда temp - первые 8 бит temp1 - 9-ый бит

TERMOUT2:

ldi ramsize,8

termout2loop:

sbi PIND,DQ

sbrc temp2,0

cbi PIND,DQ

sbi PIND,CLK

cbi PIND,CLK

ror temp2

dec ramsize

brne termout2loop ;если не закончили повторить

ldi ramsize,0x8

termout2loopdat:

clc

sbi PIND,CLK

sbic PIND,DQ

sec

cbi PIND,CLK

ror temp2

dec ramsize

brne termout2loopdat ;если не закончили повторить

ldi temp1,0

sbi PIND,CLK

sbic PIND,DQ

ldi temp1,1

cbi PIND,CLK

cbi DDRD,DQ

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\* получение кода нажатой клавиши если она нажата более 20 мс

;\* код кнопки находится в регистре key если сброшен бит Z

KEYPRESS:

ldi temp1,20 ;количество попыток для гашения

;дребезга контактов

keyloop:

in temp,PIND

andi temp,0b00111000

breq keynopress

sbic PIND,K1

ldi key,1

sbic PIND,K2

ldi key,2

sbic PIND,K3

ldi key,3

dec temp1

brne keyloop ;если не закончили повторить

keynopress:

and key,key

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\* ожидание отпускания клавиши

KEYUP:

ldi temp1,20 ;количество попыток для гашения

;дребезга контактов

keyuloop:

in temp,PIND

andi temp,0b001110000

breq unpress

dec temp1

brne KEYUP ;если не закончили повторить

unpress:

and key,key

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; программа переводит температуру в зависимости от режима

; и выводит на индикатор для индикации

; calckterm

outTmax:

mov fbinL,TMAXL

mov fbinH,TMAXH

rjmp outterm

outTmin:

mov fbinL,TMAXL

mov fbinH,TMAXH

rjmp outterm

calckterm:

mov temp,tmode

rcall STRLCD

mov fbinL,TNORML

mov fbinH,TNORMH

cpi tmode,1

breq outTmin

cpi tmode,2

breq outTmax

ldi temp,0xb

outterm:

ldi ZH,high(DLCD+5)

ldi ZL,low(DLCD+5)

ldi temp2,0xb

sbrs fbinH,0

ldi temp2,0x10

; ldi temp3,0

; sub temp3,temp2

; ror temp3

st Z+,temp2

inc ZL

inc ZL

ldi temp2,5

sbrs fbinL,0

ldi temp2,0

st Z+,temp2

ldi fbinH,0

clc

ror fbinL

rcall bin16BCD5

mov temp3,fbinH

mov temp2,fbinL

swap temp2

andi temp3,0xf

andi temp2,0xf

andi temp,0xf

rcall getcod

mov temp1,temp

set

bst temp1,4

mov temp,temp2

rcall getcod

mov temp2,temp

mov temp,temp3

rcall getcod

mov temp3,temp

ldi ZH,high(DLCD+6)

ldi ZL,low(DLCD+6)

st Z+,temp3

st Z+,temp2

st Z+,temp1

rcall LCDOUT

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; программа возвращает код символа для индикатора

; значение temp заменяется на значение необходимое для отображения на

; индикаторе и оно помещается назад в ячейку temp

getcod:

ldi ZH,high(LCDCG\*2)

ldi ZL,low(LCDCG\*2);init Z-pointer

add ZL,temp

lpm ;get constant

mov temp,r0

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; данные для отображения на индикаторе

LCDCG: ;тип код

.DB 0xEE ;число 0

.DB 0x60 ;число 1

.DB 0x2F ;число 2

.DB 0x6D ;число 3

.DB 0xE1 ;число 4

.DB 0xCD ;число 5

.DB 0xCF ;число 6

.DB 0x68 ;число 7

.DB 0xEF ;число 8

.DB 0xED ;число 9

.DB 0x10 ;символ. a

.DB 0x01 ;символ - b

.DB 0x87 ;символ t c

.DB 0x46 ;символ u d

.DB 0x67 ;символ d e

.DB 0x07 ;символ c f

.DB 0x00 ;символ blanck 10 пустой символ

;строка tc-

tcstr:

.DB 0x0c,0x0f,0x0b,0x10

;строка tu-

tustr:

.DB 0x0c,0x0d,0x0b,0x10

;cтрока td-

tdstr:

.DB 0x0c,0x0e,0x0b,0x10

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*

;\* PROGRAM RESET - главная программа

;\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

RESET:

ldi temp,low(RAMEND)

out SPL,temp ; настройка стека на верх памяти данных

rcall port\_set ; настроить порты

rcall CLRLCD

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*

;\* mainloop - главный цикл программы

;\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

mainloop:

ldi temp1,250

mov r5,temp1

mainloop1:

rcall KEYPRESS ;определения нажатия кнопки

brne keyis ; переход если хотябы одна кнопка нажата

rjmp returnin ;возврат к основному цыклу

keyis:

ldi temp1,250

mov r5,temp1

cpi temp,1

rcall key1press

cpi temp,2

rcall key2press

cpi temp,3

rcall key3press

returnin:

ldi temp1,0xaa ; чтениее текущей температуры термостата

rcall TERMIN

ldi temp2,0xaa ; инициализация термостата

rcall TERMIN

mov TNORML,temp

mov TNORMH,temp1

rcall calckterm ; расчет температуры

dec r5

brne mainloop1 ; после ожиданжжия в течении минуты

ldi tmode,0 ; возвратиться в основной режим

rjmp mainloop

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\* обработка клавиши режима

key1press:

and tmode,tmode

brne k1setmax

dec tmode

rjmp keyret

k1setmax:

ldi temp,2

mov tmode,temp

keyret:

rcall calckterm

rjmp KEYUP

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\* обработка клавиши изменения значения вверх

key2press:

cpi tmode,1

brne key2min

cpi tmode,2

brne key2max

rjmp keyret

key2min:

sbrs TMINH,0

rjmp key2minm

inc TMINL

brmi key2mins

rjmp keyret

key2mins:

ldi TMINH,0

rjmp keyret

key2minm:

cpi TMINL,0xfa

brmi keyret

inc TMINL

rjmp keyret

key2max:

sbrs TMAXH,0

rjmp key2maxm

inc TMINL

brmi key2maxs

rjmp keyret

key2maxs:

ldi TMAXH,0

rjmp keyret

key2maxm:

cpi TMAXL,0xfa

brmi keyret

inc TMAXL

rjmp keyret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\* обработка клавиши изменения значения вниз

key3press:

cpi tmode,1

brne key3min

cpi tmode,2

brne key3max

rjmp keyret

key3min:

sbrc TMINH,0

rjmp key3minm

dec TMINL

brmi key3mins

rjmp keyret

key3mins:

ldi TMINH,1

rjmp keyret

key3minm:

cpi TMINL,0x92

brmi keyret

dec TMINL

rjmp keyret

key3max:

sbrc TMAXH,0

rjmp key3maxm

dec TMAXL

brmi key3maxs

rjmp keyret

key3maxs:

ldi TMAXH,1

rjmp keyret

key3maxm:

cpi TMAXL,0x92

brmi keyret

dec TMAXL

rjmp keyret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;1. Текст программы "bin16BCD5"- преобразование 16-битного двоичного

;значения в упакованный BCD формат

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\* Количество слов кода:25 + возврат

;\* Количество циклов:25/176 (Мин/Макс) + возврат

;\* Использованные младшие регистры:нет

;\* Использованные старшие регистры:4(fbinL,fbinH/tBCD0,tBCD1,tBCD2)

;\* Использованные указатели:нет

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\* Регистровые переменные подпрограммы

.def fbinL =r16;двоичное значение, младший байт

.def fbinH =r17;двоичное значение, старший байт

.def tBCD0 =r17;BCD значение, цифры 1 и 0

.def tBCD1 =r18;BCD значение, цифры 3 и 2

.def tBCD2 =r19;BCD значение, цифра 4

;Примечание: Переменные fbinH и tBCD0 должны размещаться в одном

;регистре.

;\*\*\*\*\* Код

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

bin16BCD5:

ldi tBCD2, -1

bin16BCD5\_loop\_1:

inc tBCD2

subi fbinL, low(10000)

sbci fbinH, high(10000)

brsh bin16BCD5\_loop\_1

subi fbinL, low(-10000)

sbci fbinH, high(-10000)

ldi tBCD1, -0x11

bin16BCD5\_loop\_2:

subi tBCD1, -0x10

subi fbinL, low(1000)

sbci fbinH, high(1000)

brsh bin16BCD5\_loop\_2

subi fbinL, low(-1000)

sbci fbinH, high(-1000)

bin16BCD5\_loop\_3:

inc tBCD1

subi fbinL, low(100)

sbci fbinH, high(100)

brsh bin16BCD5\_loop\_3

subi fbinL, -100

ldi tBCD0, -0x10

bin16BCD5\_loop\_4:

subi tBCD0, -0x10

subi fbinL, 10

brsh bin16BCD5\_loop\_4

subi fbinL, -10

add tBCD0, fbinL

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*