МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН

НУКУССКИЙ ФИЛИАЛ ТАШКЕНТСКОГО УНИВЕРСИТЕТА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ИМЕНИ АЛЬ ХОРЕЗМИ

Факультет: «Компьютерный инжиниринг»

Направление: Телекоммуникационные технологии

По предмету «Микропроцессоры»

КУРСОВАЯ РАБОТА

На тему: Измеритель температуры жидкости

Выполнил(а): **Зинуллаева А.**

Принял(а): **Мамутова В.**

Нукус 2018

Содержание:

Введение

1. Анализ поставленной задачи

1.1 Аналитический обзор микроконтроллера AT90S1200

1.2 Аналитический обзор цифрового термометра

2. Проектирование принципиальной схемы устройства

2.1 Схема включения микропроцессора

2.2 Формирование тактовых импульсов

2.3 Схема сброса

2.4 Схема подключения цифрового термометра

2.5 Схема подключения ЖКИ

3. Проектирование программного обеспечения микроконтроллера

3.1 Разработка алгоритма программы

3.2 Проектирование процедур управления интерфейсными устройствами

3.3 Проектирование процедуры инициализации аппаратуры микроконтроллера

3.4 Инициализация цифрового термометра DS1620

3.5 Инициализация ЖКИ

3.6 Проектирование процедуры Main()

Заключение

Список использованной литературы

Введение

В связи с повсеместным использованием цифровых управляющих систем постоянно растет необходимость разработки и их усовершенствования.

Большинство цифровых систем строится на микропроцессорах либо на микроконтроллерах. При помощи микропроцессорных систем происходит управление различными технологическими процессами и операциями. Данные системы практически универсальны, так как они имеют очень высокое быстродействие, и достаточную разрядность для различных выполнения различных расчетов на производстве. Используя в данных системах ППЗУ, возможно, при помощи одной компьютерной системы управление различным оборудованием. То есть необходимо изменение только программы управления.

Центральное место в структуре микропроцессорного устройства занимает микропроцессор, который выполняет арифметические и логические операции над данными, программное управление процессором обработки информации, организует взаимодействие всех устройств, входящих в систему. Микропроцессор представляет собой функционально законченное устройство, состоящее из одной или нескольких программно-управляемых БИС и предназначенное для выполнения операций по обработке информации и управления вычислительным процессом.

В данной курсовой работе необходимо разработать измеритель температуры жидкости. Данное устройство будет построено на базе Clasic серии микроконтроллеров Atmel.

Датчиком температуры является цифровой термометр DS1620. Данный термометр позволяет по последовательному интерфейсу считывать показания в цифровом виде.

В качестве индикатора применен ЖКИ-модуль.

1. Анализ поставленной задачи

Темой курсового проекта является «Измеритель температуры жидкости». Для реализации поставленной задачи, нам необходимо использовать процессор AT1200S, вместо него мы будем использовать AT90S1200. Он является полным аналогом своего предшественника. Данная система позволяет очень быстро и точно производить измерение температуры.

Рассмотрим структурную схему, приведенную на рисунке 1.

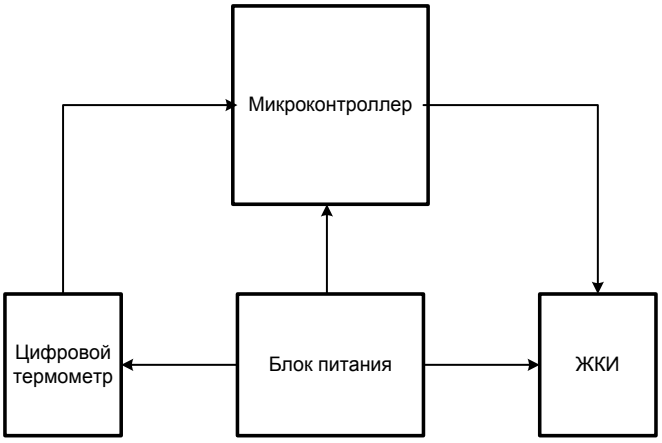

Рисунок 1 – Структурная схема устройства.

В качестве микроконтроллера по заданию задан AT90S1200. Это микроконтроллер из семейства clasic.

1.1 Аналитический обзор микроконтроллера AT90S1200

В состав данного микроконтроллера входит следующие периферийные устройства:

- встроенный сторожевой таймер;
- аналоговый компаратор.

На рисунке 2 приведен микроконтроллер AT90S1200.

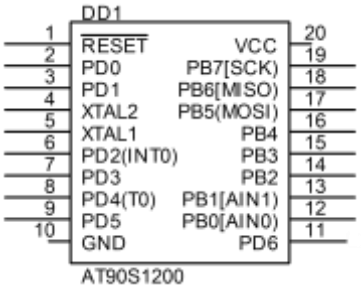

Рисунок 1.1 – микроконтроллер AT90S1200

Характеристики AT90S1200:

– Память программ 1 Кбайт
– Память данных (EEPROM) 64 байт
– Память данных (ОЗУ) -
– Количество лин. ввод/выв. 15
– Напряжение питания 2.7–5.5 В
– Тактовая частота 0 – 12 МГц

AT90S1200 является 8-ми разрядным CMOS микроконтроллером с низким энергопотреблением, основанным на усовершенствованной AVR RISC архитектуре. Благодаря выполнению высокопроизводительных инструкций за один период тактового сигнала, AT90S1200 достигает производительности, приближающейся к уровню 1 MIPS на МГц, обеспечивая разработчику возможность оптимизировать уровень энергопотребления в соответствии с необходимой вычислительной производительностью. Ядро AVR содержит мощный набор инструкций и 32 рабочих регистра общего назначения. Все 32 регистра напрямую подключены к арифметико-логическому устройству (АЛУ), что обеспечивает доступ к двум независимым регистрам при выполнении одной инструкции за один такт. В результате, данная архитектура имеет более высокую эффективность кода, при повышении пропускной способности, вплоть до 10 раз, по сравнению со стандартными микроконтроллерами CISC.

Архитектура также эффективно поддерживает языки высокого уровня, как и ультра- уплотненные программы на ассемблерном коде. AT90S1200 имеет: 1 Кбайт Flash - памяти с поддержкой внутрисистемного программирования, 64 байт EEPROM, 15 линий I/O общего назначения, 32 рабочих регистра общего назначения, внутренние и внешние прерывания, программируемый следящий таймер с встроенным тактовым генератором и программируемый последовательный порт SPI для загрузки программ, а также, два программно выбираемых режима экономии энергопотребления. Режим ожидания «Idle Mode» останавливает CPU, но позволяет функционировать регистрам, таймеру/ счетчику, следящему таймеру и системе прерываний. Режим экономии энергопотребления «Power Down» сохраняет значения регистров, но останавливает тактовый генератор, отключая все остальные функции микроконтроллера, вплоть до следующего внешнего прерывания, или до аппаратной инициализации.

Устройство производится с применением технологии энергонезависимой памяти с высокой плотностью размещения, разработанной в корпорации Atmel. Встроенная Flash - память с поддержкой внутрисистемного программирования обеспечивает возможность перепрограммирования программного кода в составе системы, посредством SPI последовательного интерфейса, или с помощью стандартного программатора энергонезависимой памяти. Благодаря совмещению усовершенствованного 8-ми разрядного RISC CPU с Flash- памятью с поддержкой внутрисистемного программирования на одном кристалле получился высокопроизводительный микроконтроллер AT90S1200, обеспечивающий гибкое и экономически- высокоэффективное решение для многих приложений встраиваемых систем управления. AVR AT90S1200 поддерживается полным набором программ и пакетов для разработки, включая: макроассемблеры, отладчики/ симуляторы программ,

внутрисхемные эмуляторы и наборы для макетирования. На рисунке 1.2 приведена внутренняя структура AT90S1200

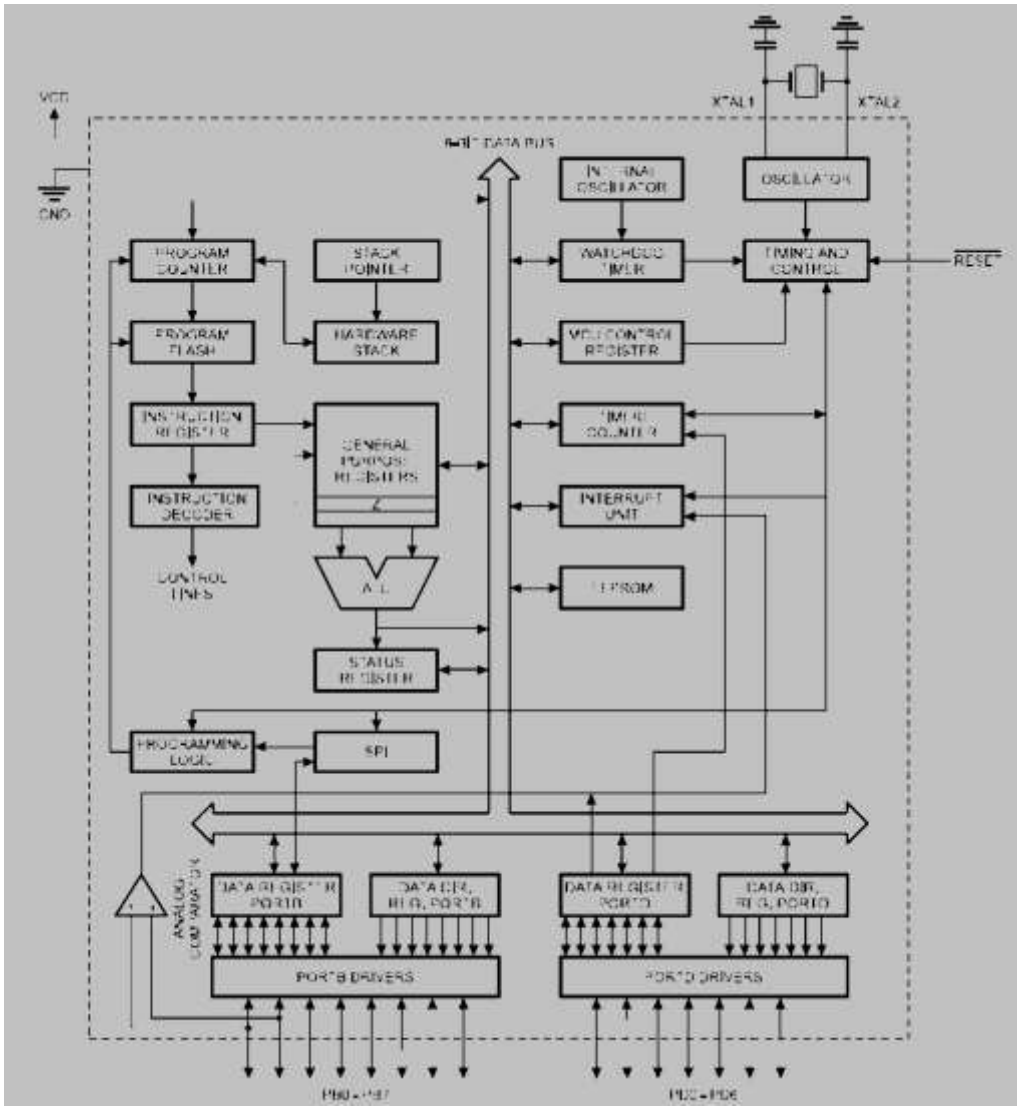

Рисунок 1.2 – Внутренняя структура AT90S1200

1.3 Аналитический обзор цифрового термометра

По заданию в качестве цифрового термометра необходимо применить цифровую микросхему DS 1620. Данная микросхема не требует внешних компонентов, напряжение питания может варьировать от 2.7 В до 5.5 В, диапазон температур -55 °С – +125 °С с точностью 0.5 °С, время измерения до 1 секунды. На рисунке 1.3 приведен цифровой термометр DS 1620.

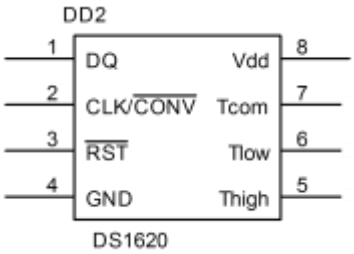

Рисунок 1.3 – Структурная схема цифрового термометра DS 1620.

Микросхема DS1620 это термометр и термостат с цифровым вводом/выводом, обеспечивающий точность ±0.5°C. При использовании в качестве термометра, данные считываются через 3-проводную последовательную шину в дополнительном 9-битном коде с ценой младшего разряда ±0.5°C. Для приложений требующих более высокого разрешения, пользователь может прочитать дополнительные регистры и произвести простые арифметические действия, чтобы достичь более чем 12-битового разрешения (с ценой самого младшего разряда 0.0625°C).

При использовании в качестве термостата, микросхема DS1620 отличается наличием во внутренней энергонезависимой памяти (EEPROM) программируемых пользователем уставок по превышению температуры (TH) и по понижению температуры (TL). Три специальных логических выхода срабатывают, когда соответствующие уставки пересекаются. Один срабатывает, когда пересекается уставка TH, другой при пересечении TL, и третья срабатывает, когда TH достигнут, и выход будет оставаться активным до тех пор, пока температура не упадёт ниже TL (программируемый гистерезис). DS1620 может быть запрограммирован с этими уставками и использоваться в автономном приложении только как термостат до тех пор, пока не понадобится их перенастроить.

Микросхема DS1620 прелагается в 300mil, 8-контактном PDIP и 208mil, 8-контактном SOIC. Для приложений, которым не требуется точность ±0.5°C, доступна микросхема DS1720 с пониженной точностью ±2.5°C, более дешёвая полностью совместимая микросхема (только в корпусе SOIC).

2. Проектирование принципиальной схемы устройства

2.1 Схема включения микроконтроллера

Микроконтроллер AT90S1200 содержит 2 портов ввода/вывода. Порт D используется для связи с цифровым термометром по 3-wire интерфейсу, а так же для управления ЖК-индикатором.

На рисунке 2.1 приведена структурная схема включения микроконтроллера.

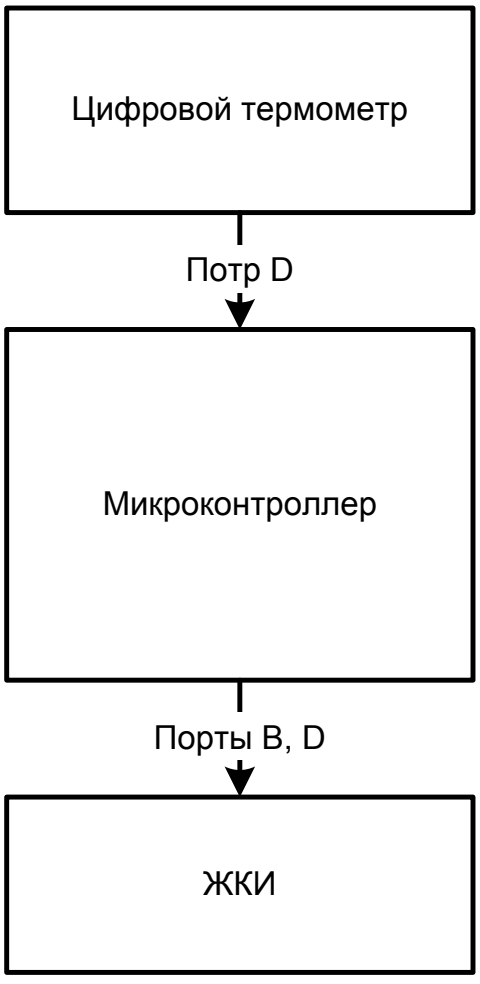

Рисунок 2.1 – Структурная схема включения микроконтроллера

2.2Формирование тактовых импульсов

Источником тактовых импульсов в микроконтроллере AT90S1200 может быть:

– Генератор с внешним резонатором;

– Генератор с внутренней RC-цепочкой;

Наиболее предпочтительным в данном блоке является генератор с внешним резонатором т. к. он позволяет задавать любую тактовую частоту, которая зависит только от кварцевого резонатора, на которой может работать микроконтроллер. Это стабильный генератор с точной выдержкой частоты генерации.

Использование внешнего генератора требует наличия дополнительной аппаратуры.

Генератор с внутренний и внешний RC-цепочкой не гарантирует стабильность частоты.

На рисунке 2.2 приведена схема включения тактового генератора.

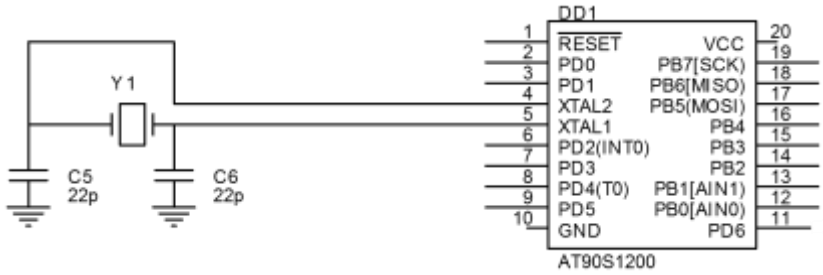

Рисунок 2.2 – Схема включения тактового генератора с внешним резонатором.

2.3. Схема сброса

На рисунке 2.3 приведена аппаратная схема сброса по включению питания. Данная схема необходима для первичной инициализации аппаратуры микроконтроллера.

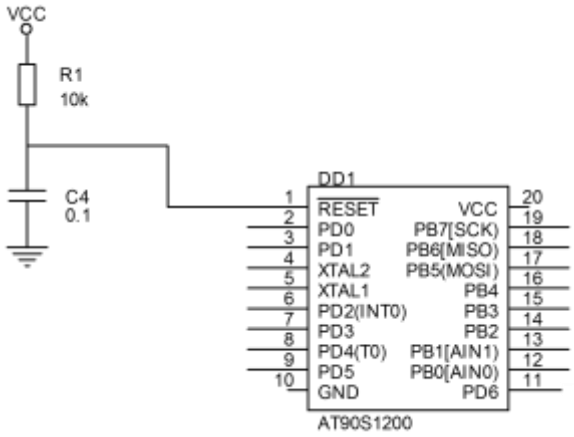

Рисунок 2.3 – Аппаратная схема сброса по включению.

2.4 Схема подключения цифрового термометра

Для подключения цифрового термометра используется три порта ввода/вывод. На рисунке 2.4 приведена схема включения термометра.

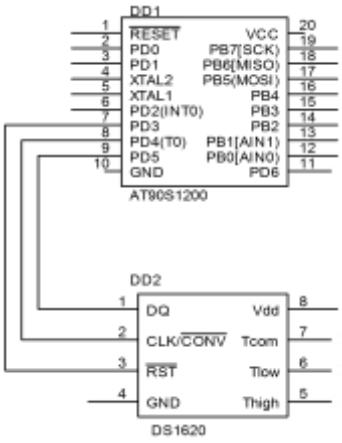

Рисунок 2.4 – Схема включения термометра

Функциональная схема приведена на рисунке 2.5

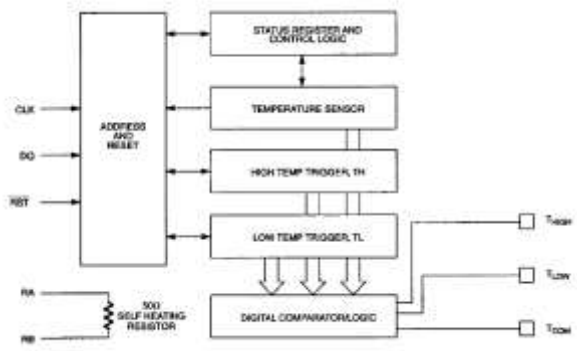

Рисунок 2.5 – Функциональная схема DS1620.

2.5 Схема подключения ЖКИ

ЖКИ подключен к микроконтроллеру AT90S1200 с помощью 8-ми разрядной шиной.

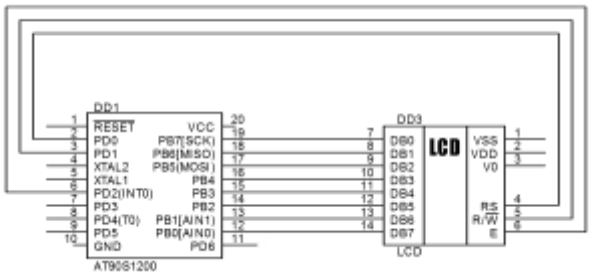

Рисунок 2.6 – Схема подключения ЖКИ.

Алфавитно-цифровые ЖКИ - модули представляют собой недорогое и удобное решение, позволяющее сэкономить время и ресурсы при разработке новых изделий, при этом обеспечивают отображение большого объема информации при хорошей различимости и низком энергопотреблении. Возможность оснащения ЖКИ - модулей задней подсветкой позволяет эксплуатировать их в условиях с пониженной или нулевой освещенностью, а исполнение с расширенным диапазоном температур (-20°C...+70°C) в сложных эксплуатационных условиях, в том числе в переносной, полевой и даже, иногда, в бортовой аппаратуре.

В соответствии с временной диаграммой в исходном состоянии сигнал E = 0, сигнал R/W = 0, значение сигнала RS - произвольное, шина данных DBO...DB7 в состоянии высокого импеданса (HI). Такое состояние управляющих сигналов (E и R/W) должно поддерживаться все время в промежутках между операциями обмена с ЖКИ-модулем. Шина данных в эти моменты в принципе свободна, и может использоваться в мультиплексном режиме для каких-либо других целей, например, для сканирования матрицы клавиатуры. Естественно, необходимо позаботиться об исключении конфликтов на шине данных в момент совершения операций обмена с ЖКИ-модулем.

Последовательности действий, которые необходимо выполнять управляющей системе при совершении операций записи и чтения для 8-ми разрядной шины приведены соответственно в таблицах 1, 2. Для нормальной работы ЖКИ необходимо сформировать временные диаграммы приведенные на рисунках 2. 7 и 2.8

Таблица 1. Операции записи для 8-ми разрядной шины

Установить значение линии RS
Вывести значение байта данных на линии шины DB0...DB7
Установить линию E = 1
Установить линию У = 0
Установить линии шины DB0...DB7 = HI

Таблица 2. Операции чтения для 8-ми разрядной шины

Установить значение линии RS
Установить линию R/W = 1
Установить линию E = 1

Считать значение байта данных с линий шины DB0...DB7

Установить линию E = 0

Установить линию R/W = 0

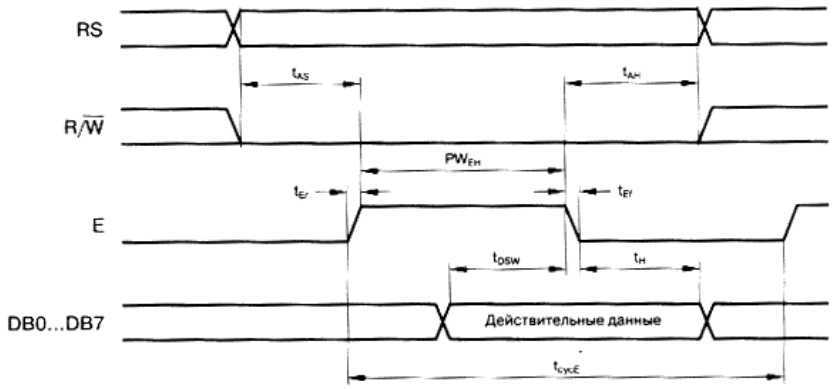

Рисунок 2.7 – Временная диаграмма операции записи

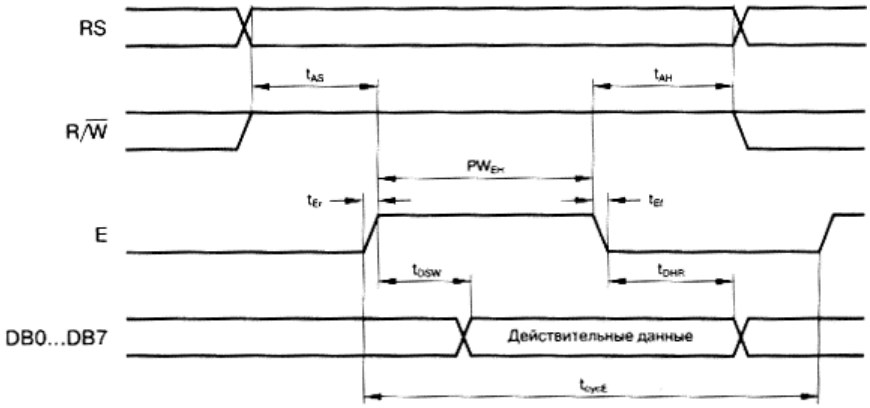

Рисунок 2.8 – Временная диаграмма операции чтения

3. Проектирование программного обеспечения микроконтроллера

3.1 Разработка алгоритма программы

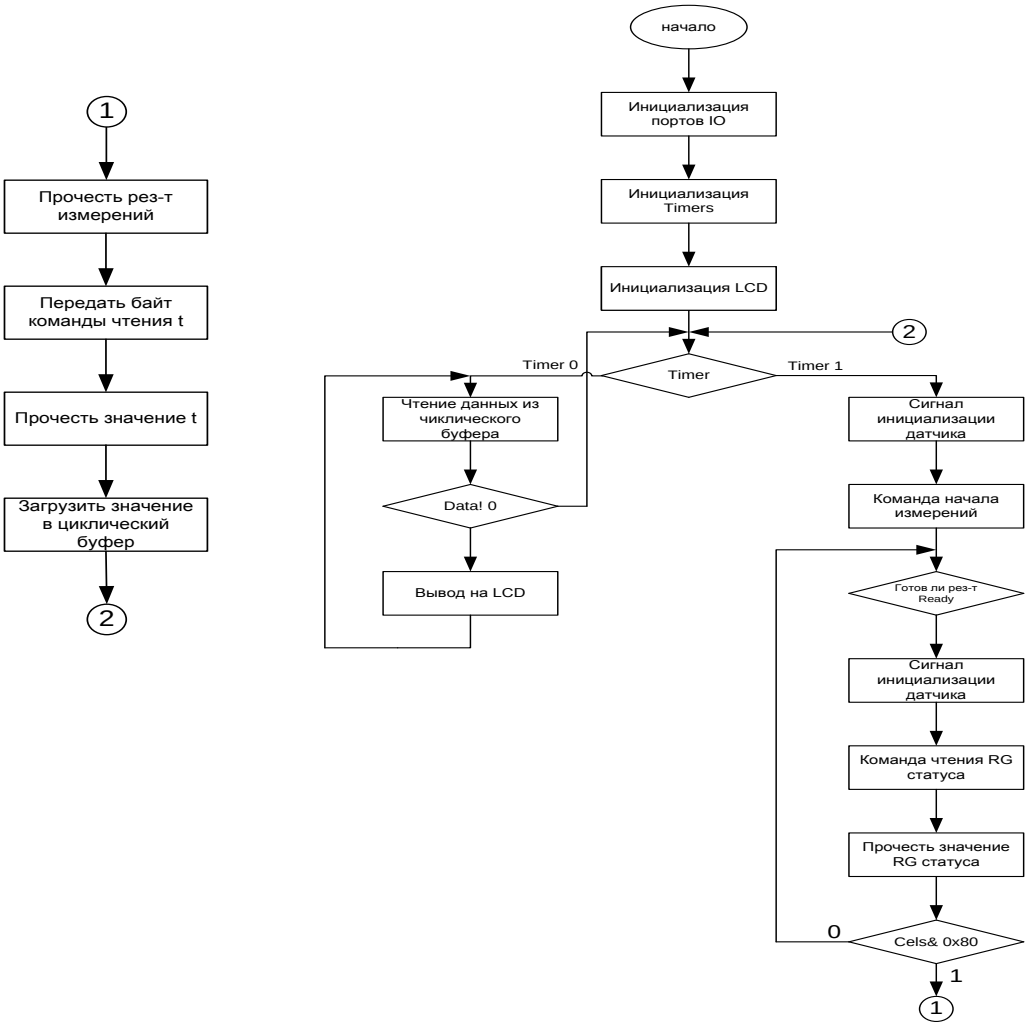

Рисунок 3.1 – Алгоритм функционирования цифрового термометра.

3.2 Проектирование процедур управления периферийными устройствами

Разрабатываемое устройство выполняет следующие операции:

3.1. Запрос текущей температуры

3.2. Обработка полученной информации.

```
#define ENABLE_BIT_DEFINITIONS
#include <tiny2313.h>
#include "ctype.h"
#include "stdlib.h"
#define PrescalerTmr0 4                    // timer0 counts clk/256
// OscFrq 7342800                          osc frequency in Hz
// OscPeriod 1/OscFrq * 1000000000 =  136.1878     osc Period in ns
//Tmr0ClkPeriod = OscPeriod*256 = 34864.07        Timer0 Clk Period in ns
//Tmr0_Interval = 1000000                  timer0 overflow interval in ns (1ms=1000000ns)
//Tmr0_Ticks = Tmr0_Interval/Tmr0ClkPeriod = 28,68 timer0 steps for 1 ms delay
#define Tmr0_Reload   256 - 29             // timer0 Reload value for 1 ms
#define TOIE0   0
//==CircularBuffer
#define CircBufLen 32
unsigned char CircBuf[CircBufLen];
unsigned char CircBufHead = 0;
unsigned char CircBufTail = 0;
//==GlobalVariables
unsigned int Var2 = 0;
//==Declare external functions
void DisplayInit(void); // Инициализация индикатора
```

```
void SendDataToDisplay(unsigned char Data, unsigned char Mode);

//==Declare internal functionsunsigned char CircBufGet(void);
void CircBufPut (unsigned char data);
//==VirtualTimerVariables
unsigned char Tmr0Flag = 0;
unsigned char TmrCnt[2];
unsigned char TmrPreLoad[2];
unsigned char TmrFlag[2]={0,0};
//0    - timer disabled
//0x01 - timer is started and counting, not reloadable
//0x81 - counting, reloadable
//0x02 - ready, stopped
//0x83 - ready, reloaded, counting
```

3.3 Проектирование процедуры инициализации аппаратуры микроконтроллера

Процедура инициализации производит настройку: портов ввода/вывода, периферийных аппаратных устройств, а так же внешних устройств которые требуют инициализации.

```
//== Port Initialisation ===============
void Init(void)
{
DDRD = 0xf0; //PD3-PD0 as input
PORTD = 0xff; //Turn ON PullUP for PortB pins
DDRB = 0xff;      //Port B pins as output
PORTB = 0x00;
}
```

```
//== Virtual Timer Initialisation ==========
void InitTimers(void)
{
#asm("cli");
TCCR0B=PrescalerTmr0;
TIMSK |= (1 << TOIE0);  //Enable Timer0 Interrupt
TCNT0=Tmr0_Reload;
TmrPreLoad[0]=250;
TmrCnt[0]=250;
TmrFlag[1]=0x81;
TmrPreLoad[1]=10;
#asm("sei");
}
//================================
char TimeDelay_us(char x)  //near 1us time delay
{
char i,j,k,n;
j=1;
for (i=0;i<x;i++)
{
k=j+1;
n=k-j;
}
return n;
}
//== Circular Buffer Write =====================
void CircBufPut (unsigned char data)
{
unsigned char tmphead;
```

```
tmphead = CircBufHead + 1;
if (tmphead>=CircBufLen)
{
tmphead=0;
}
CircBuf[tmphead] = data;
CircBufHead = tmphead;
}
//== Circular Buffer Read ==========
unsigned char CircBufGet(void)
{
unsigned char tmptail;
if (CircBufHead != CircBufTail)
{
tmptail=CircBufTail+1;
if (tmptail>=CircBufLen)
{
tmptail=0;
}
CircBufTail = tmptail;
return CircBuf[tmptail];
}
else
{
return 0;
}
}
```

3.4 Инициализация цифрового термометра DS1620

```
char DS1620Init(void)
{
char Presence;
DDRD |= 0x20;
PORTD &= ~0x20;
TimeDelay_us(200);
TimeDelay_us(200);
TimeDelay_us(200);
DDRB &= ~0x10;
PORTB |= 0x10;
TimeDelay_us(20);
Presence = PIND & 0x10;
TimeDelay_us(200);
DDRD |= 0x20;
PORTD |= 0x20;
TimeDelay_us(200);
return Presence;
}
void DS1620WriteBit(char Value)
{
#asm("cli");
DDRD |= 0x20; //output 5
PORTD &= ~0x20;
TimeDelay_us(5);
if(Value!=0)  //if data bit = H => output 5
{
PORTD |= 0x20;
```

```
}
TimeDelay_us(70);
PORTD |= 0x20; //output 5
TimeDelay_us(5);
#asm("sei");
}
void DS1620WriteByte(char data)
{
char loop, CurrentBit;
for (loop = 0; loop < 8; loop++)  // Loop to write each bit in the byte, LS-bit
first
{
CurrentBit = data & 0x01;
DS1620WriteBit(CurrentBit);
data >>= 1; // shift the data byte for the next bit
}
}
char DS1620ReadBit(void)
{
char Value;
#asm("cli");
DDRD |= 0x20; //output 5
PORTD &= ~0x20;
TimeDelay_us(5);
DDRD &= ~0x20; //input
PORTD |= 0x20;
TimeDelay_us(10);
Value = PIND & 0x20;  //read bit
TimeDelay_us(55);
```

```
DDRD |= 0x20; //output 5
PORTD |= 0x20;
TimeDelay_us(5);
#asm("sei");
return Value;
}
//==
char DS1620ReadByte(void)
{
char loop, result=0, CurrentBit;
for (loop = 0; loop < 8; loop++)
{
result >>= 1; // shift the result right to get it ready for the next bit
CurrentBit = DS1620ReadBit();
if (CurrentBit != 0) // if result is one, then set MS bit
{
result |= 0x80;
}
}
return result;
}
```

3.5 Инициализация и настройка ЖКИ

```
#include <tiny2313.h>
/*
#define LCD_E PORTC_Bit4
#define LCD_RW PORTC_Bit5
#define LCD_RS PORTC_Bit6
```

```
#define LCD_DATA PORTC
#define LCD_PIN PINC
#define LCD_DDR DDRC
*/
#define E PORTD.2
#define WR PORTD.1
#define RS PORTD.0
#define LCD_DATA PORTD
#define LCD_PIN PIND
#define LCD_DDR DDRD
#define CLRBIT(ADDR, BIT)       (ADDR |= (1<<BIT))
#define SETBIT(ADDR, BIT)       (ADDR &= ~(1<<BIT))
char
ini_cmd[]={0x03,0x03,0x03,0x02,0x02,0x0d,0x00,0x0d,0x00,0x01,0x00,0x06};
//==================================
void Delay(int i)  // программная задержка
{
while(--i>0x00);
}
//==============================
void SendDataToDisplay(unsigned char Data, unsigned char Mode)
{
//PORTB - 8bit Data
/*PORTD - PD0 - RS
PD1 - RW
PD2 - E */
CLRBIT(PORTD,E);
if (Mode)
SETBIT(PORTD,RS);
```

```
else
CLRBIT(PORTD,RS);
PORTB = Data;
CLRBIT(PORTD,WR);
SETBIT(PORTD,E);
Delay(4);
CLRBIT(PORTD,E);
}
//=======================
void DisplayInit(void)
{
Delay(30);
SendDataToDisplay(0x30,1); //режим работы дисплея – ширина шины данных 8 бит
Delay(5);
SendDataToDisplay(0x30,1);
Delay(1);
SendDataToDisplay(0x30,1);
SendDataToDisplay(0x38,1); // шина данных 8 бит
//размер развертки 2 строки
//размер матр. Символов – 5x10
SendDataToDisplay(0x08,1); //выкл. Наличие изображения
SendDataToDisplay(1,1);    //очистка экрана
SendDataToDisplay(0x6,1);  //счетчик адреса настроить на увеличение
SendDataToDisplay(0xC,1);  //вкл. изображение
}
unsigned char ReadDatafromDisplay(unsigned char Mode)
{
unsigned char a;
```

```
CLRBIT(PORTD, E);
if (Mode)
SETBIT(PORTD,RS);
else
CLRBIT(PORTD,RS);
DDRB &= 0x00; //установка порта на чтение
PORTB |= 0xFF;
Delay(4);
a = PORTB;
CLRBIT(PORTD,E);
return a;
}
```

3.6 Проектирование процедуры Main()

Процедура Main(), является основной исполняемой процедурой из которой начинается выполнение программы. Поэтому все действии нужно выполнять в этой процедуре.

В начале процедуры необходимо разместить вызовы процедур инициализации.

Опрос термометра производим постоянно в бесконечном цикле.

В остальное время отображение температуры на ЖК-индикаторе.

```
//== Main Procedure
void main(void)
{
int Cels1;
char Cels,Ready;
//  unsigned int x;
Init();
```

```
InitTimers();
DisplayInit();    // lcd.c is needed!
while (1)
{
//--Virtual timer0 is used for LCD display--
if ((TmrFlag[0] & 0x02) != 0)
{
TmrFlag[0] &= ~(0x02);
{
unsigned char data;
data=CircBufGet();
while (data != 0)
{
SendDataToDisplay(data,0);  // lcd.c is needed!
data=CircBufGet();
}
}
}
//--Virtual timer1 is used to read a keypad--
if ((TmrFlag[1] & 0x02) != 0)
{
TmrFlag[1] &= ~(0x02);
//Запустить измерение
DS1620Init();
DS1620WriteByte(0xee);  //начать преобразование
//проверить готовность измерения
Ready = 0;
while (!Ready)
{
```

```
DS1620Init();
DS1620WriteByte(0xac); //read status
Cels = DS1620ReadByte();
if(Cels & 0x80)
{
Ready = 1;
}
else
{
Ready = 0;
}
}
//прочесть результат измерения
DS1620Init();
DS1620WriteByte(0xaa); //read temperature
Cels = DS1620ReadByte();
Cels1 = (int) Cels;
CircBufPut(Cels1);
}
}
}
//==Timer0 Interrupt Routine
interrupt [TIM0_OVF] void TIMER0_OVF_interrupt(void)
{
unsigned char i;
TCNT0=Tmr0_Reload;
for (i=0;i<2;i++)
{
if((TmrFlag[i] & 0x01) != 0)    // If counting bit set - serve this timer!
```

```
{
if(TmrCnt[i]==0)              // If counter is empry - serve this overflow!
{
TmrFlag[i] |= 0x02;          // Set timer overflow bit
if(TmrFlag[i] & 0x80)        // If Reload bit is set - reload this timer!
{
TmrCnt[i]=TmrPreLoad[i];  // Reload counter
}
else
{
TmrFlag[i] &= ~0x01;      // Clear counting bit
}
}
else
{
TmrCnt[i]--;
}
}
}
}
```

Заключение

В данной курсовой работе произведено проектирование цифрового термометра с возможностью отображения температуры на ЖК индикаторе.

Данное устройство было построено на базе Clasic серии микроконтроллеров Atmel.

Датчиком температуры является цифровой термометр DS1620. Данный термометр позволяет по последовательному интерфейсу считывать показания в цифровом виде.

В качестве индикатора применен ЖКИ-модуль.

Используется процессор AT1200S, вместо него я использовала AT90S1200. Он является полным аналогом своего предшественника. Данная система позволяет очень быстро и точно производить измерение температуры.

Данное устройство обладает высокой скоростью измерения, широким диапазоном измерения температуры.

Также имеет малые габариты, вес, и обладает низким энергопотреблением.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1) В.В. Скороделов "Проектирование устройств на однокристальных микроконтроллерах с RISC-архитектурой". Ч1,Ч2, Учебное пособие.

2) Угрюмов Е. П. Проектирование элементов и узлов ЭВМ. М.: Высшая школа, 1991.

3) Шило В. Л. Популярные цифровые микросхемы. М.: Радио и связь, 1988.

4) Тули М. Справочное пособие по цифровой электронике. М.: “Энергоатомиздат”, 1990.

5) Бирюков С.А. “Применение интегральных микросхем серии ТТЛ”. М.: “Патриот”, 1992.

6) Применение интегральных микросхем в электронной вычислительной технике. Справочник (под ред. Б.И. Файзуллаева, – М, Радио и связь, 1989)

7) ГОСТ 2.102-68. Виды и комплектность конструкторской документации. М.: 1988.

8) ГОСТ 2.708-81. Правила выполнения электрических схем цифровой вычислительной техники. М.: 1988.

9) Разработка и оформление конструкторской документации. РЭА. Справочник (под ред. Э.Г. Романычевой – М.: Радио и связь, 1989)

10) Ю.В. Новиков, О.А. Калашников "Разработка устройств сопряжения". Издательство "ЭКОМ", Москва, 1998г. 355 с.