Федеральное агентство по образованию

Государственное образовательное учреждение

Высшего профессионального образования

"Московский государственный технический университет им. Н.Э. Баумана"

Калужский филиал

Факультет электроники, информатики и управления

Кафедра "Системы автоматического управления и электротехника"

(ЭИУ3-КФ)

Расчетно-пояснительная записка к курсовой работе

на тему:

"Реализация цифрового термометра на основе микроконтроллера ATmega 128 (с использовнием термодатчика DS18B20)"

по дисциплине:

Проектирование алгоритмов и программного обеспечения

Калуга, 2009

Содержание

1. Теоретическая часть

1.1 Аннотация

1.2 Постановка задачи

1.3 Общие сведения о микроконтроллерах AVR

1.4 Основные параметры AVR

1.5 Программирование микроконтроллера с использованием программы на языке С

1.6 Передача данных на Com-порт

1.7 Вывод значений на ЖКИ

1.8 Описание температурного датчика DS18B20

1.9 Подключение датчика DS18B20 с использованием порта 1-Wire

2. Конструкторская часть

2.1 Общие сведения. Функциональное назначение используемых программ

2.2 Реализация

2.3 Запуск и выполнение

3. Список используемой литературы

Приложение

## 1. Теоретическая часть

## 1.1 Аннотация

Данный курсовой проект представляет собой термометр, реализованный на микроконтроллере AVR. Устройство может выполнять следующие действия:

определение температуры окружающей среды;

отображение температуры окружающей среды на ЖК дисплее;

отображение температуры окружающей среды в com-порт;

своевременное реагирование на изменение температуры, реализованное с помощью устанавливаемой по желанию задержки опроса;

## 1.2 Постановка задачи

Используя средства языка программирования необходимо создать цифровой термометр на основе микроконтроллера ATmega 128 семейства AVR. Должны выполняться действия: измерение температуры окружающей среды, вывод значений температуры на ЖК дисплей и в com-порт.

## 1.3 Общие сведения о микроконтроллерах AVR

Микроконтроллер - компьютер на одной микросхеме. Предназначен для управления различными электронными устройствами и осуществления взаимодействия между ними в соответствии с заложенной в микроконтроллер программой. В отличие от микропроцессоров, используемых в персональных компьютерах, микроконтроллеры содержат встроенные дополнительные устройства. Эти устройства выполняют свои задачи под управлением микропроцессорного ядра микроконтроллера.

С развитием микроэлектроники микроконтроллеры AVR получили широкое распространение в нашей жизни. Это устройства, в которых объединены процессор и периферийное оборудование. Такое объединение позволяет уменьшить массу стоимость и энергопотребление проектируемого устройства.

Типичный современный микроконтроллер имеет следующие встроенные схемы: центральное процессорное устройство (ЦПУ), память программ, оперативная память данных, тактовый генератор, цепь сброса, последовательный порт, таймер, таймер, аналого-цифровой преобразователь (АЦП). Устройства памяти включают оперативную память (RAM), постоянные запоминающие устройства (ROM), перепрограммируемую ROM (EPROM), электрически перепрограммируемую ROM (EEPROM). Таймеры включают, и часы реального времени, и таймеры прерываний. Средства I/O включают последовательные порты связи, параллельные порты (I/O линии), аналого-цифровые преобразователи (A/D), цифроаналоговые преобразователи (D/A), драйверы жидкокристаллического дисплея (LCD) или драйверы вакуумного флуоресцентного дисплея (VFD). Встроенные устройства обладают повышенной надежностью, поскольку они не требуют никаких внешних электрических цепей.

## 1.4 Основные параметры AVR

Основные параметры AVR:

тактовая частота до 20 МГц;

встроенный программируемый RC-генератор, частота 1, 2, 4, 8 МГц;

Flash-ПЗУ программ, программируемое в системе, 10 000 циклов перезаписи;

Энергонезависимая память данных EEPROM (100 000 циклов перезаписи);

внутреннее ОЗУ со временем доступа 1 такт;

6 аппаратных команд умножения (для семейства mega);

развитая система адресации, оптимизированная для работы с С-компиляторами;

32 регистра общего назначения (аккумулятора);

синхронный (USART) или асинхронный (UART) (в mega64 и mega128 их по 2);

синхронный последовательный порт (SPI);

двухпроводный интерфейс TWI, совместимый с интерфейсом I2C;

многоканальный PWM 8-, 9-, 10-, 16-битный ШИМ-модулятор;

10-битный АЦП с дифференциальными входами;

программируемый коэффициент усиления перед АЦП 1, 10 и 200;

встроенный источник опорного напряжения 2,56 В;

аналоговый компаратор;

сторожевой таймер - перезагружает МК при "зависании";

настраиваемая схема задержки запуска после подачи питания;

схема слежения за уровнем напряжения питания;

JTAG-интерфейс для подключения эмулятора (в МК с объемом ПЗУ от 16кбайт);

Малое энергопотребление (менее 100мкА в активном режиме на частоте 32кГц);

датчик температуры (в ATtiny25 ATtiny45 ATtiny85).

## 1.5 Программирование микроконтроллера с использованием программы на языке С

Таблица 1. Типы данных языка С для AVR

|  |  |  |
| --- | --- | --- |
| Тип | Размер (бит)  | Диапазон значений |
| bit | 1 | 0, 1 |
| char | 8 | -128…127 |
| unsigned char | 8 | 0…255 |
| signed char | 8 | -128…127 |
| int | 16 | -32768…32767 |
| short int | 16 | -32768…32767 |
| unsigned int | 16 | 0…65535 |
| signed int | 16 | -32768…32767 |
| long int | 32 | -2147483648…2147483647 |
| unsigned long int | 32 | 0…4294967295 |
| signed long int | 32 | -2147483648…2147483647 |
| float | 32 | ±1.175e-38…±3.402e38 |
| double | 32 | ±1.175e-38…±3.402e38 |

Таблица 2. Зарезервированные слова

|  |  |  |  |
| --- | --- | --- | --- |
| break | else | int | struct |
| bit | enum | interrupt | switch |
| case | extern | long | typedef |
| char | flash | register | union |
| const | float | return | unsigned |
| continue | for | short | void |
| default | funcused | signed | volatile |
| do | goto | sizeof | while |
| double | if | sfrw |  |
| eeprom | inline | static |  |

Для вывода данных в порт используется регистр PORTx, а для ввода - PINx. Для настройки порта на выход в регистр управления необходимо записать единицу, и наоборот, для настройки на вход-ноль.

## 1.6 Передача данных на Com-порт

Для связи ПК и вашего устройства необходим какой-либо физический интерфейс. Очень удобен и прост интерфейс RS232 - в ПК он реализован в виде COM-портов.

Универсальный внешний последовательный интерфейс - *CОМ-порт (*Communications Port - коммуникационный порт) присутствует в PC начиная с первых моделей. Этот порт обеспечивает *асинхронный* обмен по стандарту RS-232C. СОМ-порты реализуются на микросхемах универсальных асинхронных приемопередатчиков (*UART),* совместимых с семейством *i8250/16450/16550.* Они занимают в пространстве ввода-вывода по 8 смежных 8-битных регистров и могут располагаться по стандартным *базовым адресам* 3F8h (СОМ1), 2F8h (COM2), 3E8h (COM3), 2E8h (COM4). Порты могут вырабатывать *аппаратные прерывания* IRQ4 (обычно используются для СОМ1 и COM3) и IRQ3 (для COM2 и COM4). С внешней стороны порты имеют линии последовательных данных передачи и приема, а также набор сигналов управления и состояния, соответствующий стандарту RS-232C. СОМ-порты имеют внешние разъемы-вилки (male - "папа") *DB25P* или *DB9P,* выведенные на заднюю панель компьютера. Характерной особенностью интерфейса является применение "не ТТЛ" сигналов *-* все внешние сигналы порта двуполярные.

Гальваническая развязка отсутствует - схемная земля подключаемого устройства соединяется со схемной землей компьютера. Скорость передачи данных может достигать 115 200 бит/с.

Название порта указывает на его основное назначение - подключение коммуникационного оборудования (например, модема) для связи с другими компьютерами, сетями и периферийными устройствами. К порту могут непосредственно подключаться и периферийные устройств с последовательным интерфейсом: принтеры, плоттеры, терминалы и другие. СОМ-порт широко используется для подключения мыши, а также организации непосредственной связи двух компьютеров. К СОМ-порту подключают и электронные ключи.

Напряжение на выходе данных из COM-порта ПК (это ножка 3 в 9-ти штырьковом разъеме) по стандарту RS232 таково:

*логическая "1" от - 5 до - 15 вольт*

*логический "0" от +5 до +15 вольт*

На самом деле сигналы в COM называются по-другому! Здесь называется "логический" по аналогии названия в МК. Для согласования напряжений и логических уровней COM-порта и USART МК используют микросхемы типа *MAX232*.

*По линии TxD данные передаются от МК в ПК (через MAX232).*

*По линии RxD МК принимает данные от ПК.*

Для отправки и приема данных можно использовать гипертерминал Windows или более удобные программы (мы будем использовать Терминал из состава компилятора CodeVisionAVR).

Формат представления данных в лабораторной работы мы оставим по умолчанию: 8 Data, 1 Stop, No Parity (8-битовые данные, без проверки четности, 1 стоповый бит), а скорость выберем произвольно. В этом случае на передачу одного символа (в формате 8N1 - установленном нами выше при настройке UART) тратится время в Сек:

10 разделить на скорость USART в бод/сек.

## 1.7 Вывод значений на 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°С... +70°С) в сложных эксплуатационных условиях, в том числе в переносной, полевой и даже, иногда, в бортовой аппаратуре.

Контроллер HD44780 потенциально может управлять 2-мя строками по 40 символов в каждой (для модулей с 4-мя строками по 40 символов используются два однотипных контроллера), при матрице символа 5 х 7 точек. Контроллера также поддерживает символы с матрицей 5 х 10 точек, но в последние годы ЖКИ-модули с такой матрицей практически не встречаются, поэтому можно считать, что фактически бывают только символы 5х7 точек.

Существует несколько различных более-менее стандартных форматов ЖКИ-модулей (символов х строк): 8х2, 16х1, 16х2, 16х4, 20х1, 20х2, 20х4, 24х2, 40х2, 40х4. Встречаются и менее распространенные форматы: 8х1, 12х2, 32х2 и др., - принципиальных ограничений на комбинации и количество отображаемых символов контроллер не накладывает - модуль может иметь любое количество символов от 1 до 80, хотя в некоторых комбинациях программная адресация символов может оказаться не очень удобной.

Жидкокристаллический модуль MT-16S2H состоит из БИС контроллера управления и ЖК панели. Контроллер управления КБ1013ВГ6, производства ОАО "АНГСТРЕМ" (www.angstrem.ru), аналогичен HD44780 фирмы HITACHI и KS0066 фирмы SAMSUNG.

Модуль выпускается со светодиодной подсветкой. Внешний вид приведен на рисунке 1. Модуль позволяет отображать 1 строку из 16 символов. Символы отображаются в матрице 5х8 точек. Между символами имеются интервалы шириной в одну отображаемую точку.

Каждому отображаемому на ЖКИ символу соответствует его код в ячейке ОЗУ модуля.

Модуль содержит два вида памяти - кодов отображаемых символов и пользовательского знакогенератора, а также логику для управления ЖК панелью.

Модуль позволяет:

модуль имеет программно-переключаемые две страницы встроенного знакогенератора (алфавиты: русский, украинский, белорусский, казахский и английский)

работать как по 8-ми, так и по 4-х битной шине данных (задается при инициализации);

принимать команды с шины данных (перечень команд приведен в таблице 4);

записывать данные в ОЗУ с шины данных;

читать данные из ОЗУ на шину данных;

читать статус состояния на шину данных;

запоминать до 8-ми изображений символов, задаваемых пользователем;

выводить мигающий (или не мигающий) курсор двух типов;

управлять контрастностью и подсветкой;

Программирование и управление:

Перед началом рассмотрения принципов управления ЖКИ-модулем, обратимся к внутренней структуре контроллера HD44780, чтобы понять основные принципы построения ЖКИ-модулей на его основе. Эта информация позволит понять способы организации модулей различных форматов с точки зрения программной модели, а также мотивации конструкторов ЖКИ-модулей.

Другие элементы не являются объектом прямого взаимодействия с управляющей программой - они участвуют в процессе регенерации изображения на ЖКИ: знакогенератор, формирователь курсора, сдвиговые регистры и драйверы (напоминаем, что приведенная схема - упрощенная, и многие не важные для получения общей картины промежуточные элементы на ней опущены).

Управление контроллером ведется посредством интерфейса управляющей системы. Основными объектами взаимодействия являются регистры DR и IR. Выбор адресуемого регистра производится линией RS, если RS = 0 - адресуется регистр команд (IR), если RS = 1 - регистр данных (DR).

Данные через регистр DR, в зависимости от текущего режима, могут помещаться (или прочитываться) в видеопамять (DDRAM) или в ОЗУ знакогенератора (CGRAM) по текущему адресу, указываемому счетчиком адреса (АС). Информация, попадающая в регистр IR, интерпретируется устройством выполнения команд как управляющая последовательность. Прочтение регистра IR возвращает в 7-ми младших разрядах текущее значение счетчика АС, а в старшем разряде флаг занятости (BF).

Видеопамять, имеющая общий объем 80 байтов, предназначена для хранения кодов символов, отображаемых на ЖКИ. Видеопамять организована в две строки по 40 символов в каждой. Эта привязка является жесткой и не подлежит изменению. Другими словами, независимо от того, сколько реальных строк будет иметь каждый конкретный ЖКИ-модуль, скажем, 80 х 1 или 20 х 4, адресация видеопамяти всегда производится как к двум строкам по 40 символов.

Будучи устройством с динамической индикацией, контроллер циклически производит обновление информации на ЖКИ. Сам ЖКИ организован как матрица, состоящая в зависимости от режима работы из 8-ми (одна строка символов 5 х 7 точек), 11-ти (одна строка символов 5 х 10 точек) или 16-ти (две строки символов 5 х 7 точек) строк по 200 сегментов (когда строка насчитывает 40 символов) в каждой. Собственный драйвер конроллера HD44780 имеет только 40 выходов (SEG1... SEG40) и самостоятельно может поддерживать только 8-ми символьные ЖКИ. Это означает, что ЖКИ-модули форматов до 8 х 2 реализованы на одной единственной микросхеме HD44780, модули, имеющие большее количество символов, содержат дополнительные микросхемы драйверов, например, HD44100, каждая из которых дополнительно предоставляет управление еще 40-ка сегментами.

У контроллера HD44780 существует набор внутренних флагов, определяющих режимы работы различных элементов контроллера (таблица 7). В таблице 8 приведены значения управляющих флагов непосредственно после подачи на ЖКИ-модуль напряжения питания. Переопределение значений флагов производится специальными командами, записываемыми в регистр IR, при этом комбинации старших битов определяют группу флагов или команду, а младшие содержат собственно флаги.

Подключение модуля LCD:

Функции работы с LCD используются для интерфейса между программами на C и LCD модулями, построенными на микросхемах Hitachi HD44780 или аналогичных. Прототипы для этих функций находятся в файле lcd. h.

До включения файла lcd. h, вы должны объявить порт микроконтроллера, который будет использоваться с модулем LCD. Поддерживаются следующие форматы LCD: 1x8, 2x12, 3x12, 1x16, 2x16, 2x20, 4x20, 2x24 и 2x40 символов.

## 1.8 Описание температурного датчика DS18B20

DS18B20 - Калиброванный цифровой термометр с однопроводным 1-Wire-интерфейсом и перестраиваемой разрядностью преобразования. Диапазон измеряемых температур от - 55°C до +125°C. Считываемый с микросхемы цифровой код является результатом непосредственного прямого измерения температуры и не нуждается в дополнительных преобразованиях. Программируемая пользователем разрешающая способность встроенного АЦП может быть изменена в диапазоне от 9 до 12 разрядов выходного кода. Абсолютная погрешность преобразования меньше 0,5°C в диапазоне контролируемых температур - 10°C до +85°C. Максимальное время полного 12-ти разрядного преобразования ~750 мс. Энергонезависимая память температурных уставок микросхемы обеспечивает запись произвольных значений верхнего и нижнего контрольных порогов. Кроме того, термометр содержит встроенный логический механизм приоритетной сигнализации в 1-Wire-линию о факте выхода контролируемой им температуры за один из выбранных порогов. Узел 1-Wire-интерфейса компонента организован таким образом, что существует теоретическая возможность адресации неограниченного количества подобных устройств на одной 1-Wire-линии. Термометр имеет индивидуальный 64-разрядный регистрационный номер (групповой код 028Н) и обеспечивает возможность работы без внешнего источника энергии, только за счет паразитного питания 1-Wire-линии. Питание микросхемы через отдельный внешний вывод производится напряжением от 3,0 В до 5,5 В. Термометр размещается в транзисторном корпусе TO-92, или в 8-контактном корпусе SO для поверхностного монтажа (DS18B20Z), или 8-контактном корпусе микро-SOP для поверхностного монтажа (DS18B20U).

Выпускается специальная модификация цифрового термометра, предназначенная только для работы в режиме паразитного питания DS18B20-PAR. Она размещается в транзисторном корпусе TO-92, один из выводов которого не используется.

## 1.9 Подключение датчика DS18B20 с использованием порта 1-Wire

Однопроводной интерфейс 1-Wire, разработанный в конце 90-х годов фирмой Dallas Semiconductor, регламентирован разработчиками для применения в четырех основных сферах-приложениях:

приборы в специальных корпусах MicroCAN для решения проблем идентификации, переноса или преобразования информации (технология iButton),

программирование встроенной памяти интегральных компонентов,

идентификация элементов оборудования и защита доступа к ресурсам электронной аппаратуры,

системы автоматизации (технология сетей 1-Wire-сетей).

Первое из этих направлений широко известно на мировом рынке и уже давно пользуется заслуженной популярностью. Второе с успехом обеспечивает возможность легкой перестройки функций полупроводниковых компонентов, производимых фирмой Dallas Semiconductor и имеющих малое количество внешних выводов. Третье позволяет обеспечить недорогую, но достаточно эффективную идентификацию и надежную защиту самого разнообразного оборудования. Что касается четвертого применения, то реализация локальных распределенных систем на базе 1-Wire-шины является на сегодня де-факто наиболее оптимальным решением для большинства практических задач автоматизации. В настоящее время Dallas Semiconductor поставляет широкую номенклатуру однопроводных компонентов различных функциональных назначений для реализации самых разнообразных сетевых приложений. Поэтому имеется огромное число конкретных примеров использования 1-Wire-интерфейса для целей автоматизации в самых различных областях, и все больше разработчиков проявляют интерес к этой технологии.

Преимущества 1-Wire-технологии:

простое и оригинальное решение адресуемости абонентов,

несложный протокол,

простая структура линии связи,

малое потребление компонентов,

легкое изменение конфигурации сети,

значительная протяженность линий связи,

исключительная дешевизна всей технологии в целом.

Ведомые однопроводные компоненты, содержащие 1-Wire-интерфейс, выпускаются в двух различных видах. Либо в корпусах MicroCAN, похожих внешне на дисковый металлический аккумулятор, либо в обычных корпусах для монтажа на печатную плату.

Однако наиболее популярными ведомыми компонентами 1-Wire, на базе которых реализовано, пожалуй, наибольшее количество однопроводных приложений, безусловно, являются цифровые термометры типа DS1820. Преимущества этих цифровых термометров с точки зрения организации магистрали, по сравнению с любыми другими интегральными температурными сенсорами, а также неплохие метрологические характеристики и хорошая помехоустойчивость, уже на протяжении полутора десятков лет неизменно выводят их на первое место при построении многоточечных систем температурного контроля в диапазоне от - 55°С до125°С. Они позволяют не только осуществлять непосредственный мониторинг температуры в режиме реального времени, но и благодаря наличию встроенной энергонезависимой памяти температурных уставок, могут обеспечивать приоритетную оперативную сигнализацию в 1-Wire-линию о факте выхода контролируемого параметра за пределы заданных значений. Также поставляются более совершенные термометры DS18В20, у которых скорость преобразования определяется разрядностью результата, программируемой непосредственно по 1-Wire-линии. Цифровой код, считываемый с такого термометра, является прямым результатом измеренного значения температуры и не нуждается в дополнительных преобразованиях. Российской фирмой Rainbow Technologies получен сертификат Госстандарта России об утверждении однопроводных цифровых термометров DS1822, DS18B20, DS18S20, DS1920, производимых концерном Dallas/Maxim Integrated Products, в состав которого входит фирма Dallas Semiconductor, в качестве средств измерения. В подтверждение этого факта имеется документ о том, что данные типы приборов зарегистрированы в Государственном реестре средств измерений и допущены к применению в Российской Федерации.

## 2. Конструкторская часть

## 2.1 Общие сведения. Функциональное назначение используемых программ

Для выполнения курсовой работы были использованы следующие программные средства: компилятор языка С для микроконтроллеров AVR (CodeVisionAVR), генератор начального кода программы CodeWizard AVR с необходимой конфигурацией периферии МК (AVR Studio).

CodeVision AVR представляет собой кросс-компилятор языка С, графическую оболочку и автоматический генератор шаблонов программ, ориентированные на работу с семейством микроконтроллеров AVR фирмы Atmel.

Программа представляет собой 32-разрядное приложение для работы в операционных системах Windows 95, 98, Me, NT4.0, 2000, XP.

Кросс-компилятор включает в себя практически все элементы, соответствующие стандарту ANSI. Кроме того, в компилятор включены дополнительные возможности, ориентированные на использование архитектурных особенностей микроконтроллеров AVR и встроенных систем в целом.

Объектные файлы COFF позволяют осуществлять отладку программ с просмотром содержимого переменных. Для этого следует применять свободно распространяемый фирмой Atmel (www.atmel.com) отладчик AVR Studio.

Для отладки систем, использующих последовательную передачу данных, в графической оболочке имеется встроенная программа Terminal.

Кроме стандартных библиотек языка С, компилятор имеет библиотеки для работы с:

ЖКИ индикаторами со встроенным контроллером;

шиной I2C фирмы Philips;

датчиком температуры LM75 фирмы National Semiconductor;

часами реального времени PC8536 и PC8583 фирмы Philips, DS1302 и DS1307 фирмы Dallas Semiconductor;

однопроводным протоколом фирмы Dallas Semiconductor;

датчиками температуры DS1820 и DS1822 фирмы Dallas Semiconductor;

датчиком температуры/термостатом DS1621 фирмы Dallas Semiconductor;

памятью EEPROM DS2430 и DS2433 фирмы Dallas Semiconductor;

шиной SPI;

управлением режимами пониженного потребления энергии;

временными задержками;

преобразованием кодов Грэя.

В CodeVision AVR имеется автоматический генератор шаблонов программ, который позволяет в течение считанных минут получить готовый код для следующих функций:

настройка доступа к внешней памяти;

определение источника прерывания Reset;

инициализация портов ввода/вывода;

инициализация внешних прерываний;

инициализация таймеров/счетчиков;

инициализация сторожевого таймера;

инициализация UART;

инициализация аналогового компаратора;

инициализация встроенного АЦП;

инициализация интерфейса SPI;

инициализация поддерживаемых библиотеками CodeVision AVR микросхем, работающих с однопроводным интерфейсом и шиной I2C;

инициализация модуля ЖКИ со встроенным контроллером.

Кроме того, среда CodeVision AVR включает в себя программное обеспечение для работы с различными AVR-программаторами. После компиляции исходной программы на языке С полученный код может быть сразу записан в память программ микроконтроллера.

## 2.2 Реализация

Основные компоненты:

1. Микроконтроллер ATmega128

2. Датчик температура DS18B20

3. ЖКИ

Выделим следующие этапы в разработке программного обеспечения термометра:

1. Получение данных с датчика.

2. Передача данных на ЖКИ.

3. Обработка прерывания.

4. Передача данных на ЖКИ.

1. Получение данных с датчика:

Данные получаем с температурного датчика DS18B20, подключенного на порт В.

2. Передача данных на ЖКИ:

Данные выводим на ЖК индикатор, подключенный на порт А.

3. Обработка прерывания:

Промежуток между измерениями 500 мс.

В процессе разработки проекта возникли следующие вопросы:

1. Каким образом подключить датчик к шине 1-Wire?

2. Как зашифрована температура, передаваемая датчиком?

3. Какие специализированные команды необходимо применять для опроса датчика?

## 2.3 Запуск и выполнение

Скомпилированная программа представляет собой файл типа cof, который прошиваем на микроконтроллер ATmega 128 семейства AVR.

Соединяем компоненты рабочей модели прибора, получаем сигнал с температурного датчика, считываем температуру окружающей среды.

## 3. Список используемой литературы

1. Белов А.В. Конструирование устройств на микроконтроллерах. - СПб.: Наука и Техника, 2005. - 256 С.
2. Бородин В.Б., Калинин А.В. Системы на микроконтроллерах и БИС программируемой логики - М.: Издательство ЭКОМ, 2002. - 400 с.: илл.
3. Гребнев В.В. Микроконтроллеры семейства AVR фирмы Atmel М.: ИП РадиоСофт, 2002. - 176 с.
4. Евстифеев А.В. Микроконтроллеры AVR семейств Tiny и Mega фирмы "ATMEL" - М.: Издательский дом "Додека-ХХI", 2004. - 560 с.
5. Шпак Ю.А. Программирование на языке Си для AVR и PIC микроконтроллеров. - К: ”МК-Пресс", 2006. - 400 С.

## Приложение

*Код программы*

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This program was produced by the

CodeWizardAVR V1.25.8 Professional

Automatic Program Generator

© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s. r. l.

http://www.hpinfotech.com

Project: Digital Thermometer

Version: 1

Date: 24.11.2009

Author: Marina

Company:

Comments:

Chip type: ATmega128

Program type: Application

Clock frequency: 7,000000 MHz

Memory model: Small

External SRAM size: 0

Data Stack size: 1024

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <mega128. h>

#include <delay. h>

#include <stdio. h>

#include <lcd. h> // Alphanumeric LCD Module functions

#asm // сообщаем куда подключен датчик

. equ \_\_w1\_port=0x1B; PORTA

. equ \_\_w1\_bit=0

#endasm

 // сообщаем куда подключён ЖКИ

#asm

. equ \_\_lcd\_port=0x15; PORTC

#endasm

#include <1wire. h> // 1 Wire Bus functions

#include <ds18b20. h>

/\* >>>>>>>>>>>>>>>>maximum number of DS18B20 connected to the 1 Wire bus \*/

#define MAX\_DEVICES 8

 // Declare your global variables here

/\* >>>>>>>>>>>>>>DS18B20 devices ROM code storage area \*/

unsigned char devices;

unsigned char rom\_code [MAX\_DEVICES] [9] ;

/\*>>>>>>>>>\*/char lcd\_buffer [33] ;

void main (void)

{ int temp;

unsigned int a;

double d;

UCSR0A=0x00;

UCSR0B=0x10;

UCSR0C=0x06;

UBRR0H=0x00;

UBRR0L=0x2D;

 // >>>>>>>>>>>>>>>> Declare your local variables here

 // Declare your local variables here

 // Input/Output Ports initialization

 // Port A initialization

 // Func7=Out Func6= Out Func5= Out Func4= Out Func3= Out Func2= Out Func1= Out Func0= Out

 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T

PORTA=0x00;

DDRA=0xFF;

 // Port B initialization

 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=Out Func0=In

 // State7=T State6=T State5=T State4=T State3=T State2=T State1=0 State0=T

PORTB=0x00;

DDRB=0x02;

 // Port C initialization

 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In

 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T

PORTC=0x00;

DDRC=0x00;

 // Port D initialization

 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In

 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T

PORTD=0x00;

DDRD=0x00;

 // Port E initialization

 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In

 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T

PORTE=0x00;

DDRE=0x00;

 // Port F initialization

 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In

 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T

PORTF=0x00;

DDRF=0x00;

 // Port G initialization

 // Func4=In Func3=In Func2=In Func1=In Func0=In

 // State4=T State3=T State2=T State1=T State0=T

PORTG=0x00;

DDRG=0x00;

 // Timer/Counter 0 initialization

 // Clock source: System Clock

 // Clock value: Timer 0 Stopped

 // Mode: Normal top=FFh

 // OC0 output: Disconnected

ASSR=0x00;

TCCR0=0x00;

TCNT0=0x00;

OCR0=0x00;

 // Timer/Counter 1 initialization

 // Clock source: System Clock

 // Clock value: Timer 1 Stopped

 // Mode: Normal top=FFFFh

 // OC1A output: Discon.

 // OC1B output: Discon.

 // OC1C output: Discon.

 // Noise Canceler: Off

 // Input Capture on Falling Edge

 // Timer 1 Overflow Interrupt: Off

 // Input Capture Interrupt: Off

 // Compare A Match Interrupt: Off

 // Compare B Match Interrupt: Off

 // Compare C Match Interrupt: Off

TCCR1A=0x00;

TCCR1B=0x00;

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;

OCR1BL=0x00;

OCR1CH=0x00;

OCR1CL=0x00;

 // Timer/Counter 2 initialization

 // Clock source: System Clock

 // Clock value: Timer 2 Stopped

 // Mode: Normal top=FFh

 // OC2 output: Disconnected

TCCR2=0x00;

TCNT2=0x00;

OCR2=0x00;

 // Timer/Counter 3 initialization

 // Clock source: System Clock

 // Clock value: Timer 3 Stopped

 // Mode: Normal top=FFFFh

 // Noise Canceler: Off

 // Input Capture on Falling Edge

 // OC3A output: Discon.

 // OC3B output: Discon.

 // OC3C output: Discon.

 // Timer 3 Overflow Interrupt: Off

 // Input Capture Interrupt: Off

 // Compare A Match Interrupt: Off

 // Compare B Match Interrupt: Off

 // Compare C Match Interrupt: Off

TCCR3A=0x00;

TCCR3B=0x00;

TCNT3H=0x00;

TCNT3L=0x00;

ICR3H=0x00;

ICR3L=0x00;

OCR3AH=0x00;

OCR3AL=0x00;

OCR3BH=0x00;

OCR3BL=0x00;

OCR3CH=0x00;

OCR3CL=0x00;

 // External Interrupt (s) initialization

 // INT0: Off

 // INT1: Off

 // INT2: Off

 // INT3: Off

 // INT4: Off

 // INT5: Off

 // INT6: Off

 // INT7: Off

EICRA=0x00;

EICRB=0x00;

EIMSK=0x00;

 // Timer (s) /Counter (s) Interrupt (s) initialization

TIMSK=0x00;

ETIMSK=0x00;

 // USART0 initialization

 // Communication Parameters: 8 Data, 1 Stop, No Parity

 // USART0 Receiver: On

 // USART0 Transmitter: Off

 // USART0 Mode: Asynchronous

 // USART0 Baud Rate: 57600

UCSR0A=0x00;

UCSR0B=0x18;

UCSR0C=0x06;

UBRR0H=0x00;

UBRR0L=0x07;

 // Analog Comparator initialization

 // Analog Comparator: Off

 // Analog Comparator Input Capture by Timer/Counter 1: Off

ACSR=0x80;

SFIOR=0x00;

PORTB=0x00;

DDRB=0x01;

 // LCD module initialization

lcd\_init (16);

w1\_init ();

/\*>>>>>>>>>>>>>\*/

delay\_ms (250);

lcd\_clear ();

/\* detecting device \*/

devices=w1\_search (0xF0,rom\_code);

delay\_ms (250);

sprintf (lcd\_buffer,"%u DS18B20\nDevice detected", devices);

lcd\_puts (lcd\_buffer);

delay\_ms (250);

lcd\_clear ();

while (1)

{

temp=ds18b20\_temperature (0); // чтение температуры с датчика

if (temp>1000) { // преобразование отрицательной температуры

temp=4096-temp;

temp=-temp;

}

sprintf (lcd\_buffer,"t=%i.%u\xdfC",temp,temp%1); // запись температуры в массив для вывода на экран

lcd\_clear (); // очистка экрана

lcd\_puts (lcd\_buffer); // вывод температуры

delay\_ms (500); // ожидание 500 мс перед следующим измерением

a = ADCW;

d = (double) a / 1024 \* 5;

printf ("%f\r", a);

};

}