Министерство образования и науки Украины

Харьковский национальный университет радиоэлектроники

Кафедра:

Курсовая работа

По курсу: “ Цифровые устройства и микропроцессоры”.

По теме: “Цифровой термометр на микропроцессоре AVR - MEGA 128”.

Выполнил : Проверил:

ст. гр. ЛОЭТ – 06-1 Аврунин О.Г

Нестеренко А.А.

Зач.кн. N7211109

Харьков

2009 г.

Техническое задание

Разработать программу для микроконтроллера АVR-Mega128, выполняющую измерение температуры (с помощью датчика температуры TMP-35) в режиме непрерывного преобразования (делитель частоты равен 128) и формирующую сигналы предупреждения с помощью блока светодиодов, если значение температуры выходит за рамки диапазона 300С<Т<400С.

Реферат

Пояснительная записка содержит: 29 страниц текста, 7 рисунков, список использованных материалов источников из 6 наименований.

Цель проекта – разработка программного продукта для измерения температуры на базе микроконтроллера АVR-Mega128.

В данной работе была разработана программа на базе микроконтроллера АVR-Mega128 для измерения температуры с помощью датчика ТМР-35. Измерения температуры производятся непрерывно. Выход значений за пределы измерения сопровождается световим сигналом.

Ключевые слова: ДАТЧИК ТЕМПЕРАТУРЫ, МИКРОКОНТРОЛЛЕР, ЖКИ, ЦИФРОВОЙ ТЕРМОМЕТР, АЛГОРИТМ РАБОТЫ.

Содержание

Список используемых сокращений

Введение

1. Структура микроконтроллера ATMega128

1.1 Назначение выводов

1.2 Организация памяти и портов ввода/вывода микроконтроллера

2. Синтез структурной схемы

3. Синтез схемы электрической принципиальной цифрового термометра

4. Разработка алгоритма работы микроконтроллера

Заключение

Список источников информации

Примечание А (Текст программы работы микроконтроллера)

Список используемых сокращений

МК – микроконтроллер;

ОЗУ – оперативное запоминающее устройство;

ПЗУ – постоянное запоминающее устройство;

ПЭВМ – персональная электронно-вычислительная машина;

ЖКИ - жидко-кристаллический индикатор;

Д – датчик температуры.

Введение

Микроконтроллеры (МК) являються серцем многих современных устройств и приборов. Самой главной особенностью МК, с точки зрения конструктора-проектировщика, является то, что с их помощью легче и зачастую гораздо дешевле реализовать различные схемы [1- ст. 5].

МК может управлять различными устройствами и принимать от них данные при минимуме дополнительных узлов, так как большое число переферийных схем уже имеется непосредственно на кристалле МК. Это позволяет уменьшить размеры конструкции и снизить потребление от источника питания [1- ст. 6].

AVR - это семейство 8-разрядных RISC-микроконтроллеров фирмы Atmel. Эти МК позволяют решить множество задач встроенных систем. Они отличаються от других распостраненных в наше время микроконтроллеров большей скоростью работы, большей универсальностью. Кроме того, они очень легко программируються. Их можно перепрограммировать до 1000 раз, причем непосредственно в собранной схеме [1- ст. 9].

Имеются 3 подсемейства МК AVR:

Tiny AVR- недорогие миниатюрные МК в 8-выводном исполнении;

Classic AVR- основная линия МК с производительностью отдельных модификаций до 16 MISP, FLASH- памятью программ 2…8 Кб, памятью данных EEPROM 64...512 байт, оперативной памятью данных SRAM 128…512 байт;

Mega AVR- с производительностью 4…16 MISP для сложных приложений требующих большого обьема памяти, FLASH- памятью программ до 128 Кб, памятью данных EEPROM 64...512 байт, оперативной памятью данных SRAM 2…4 байт, встроенным 10-разрядным 8-канальным АЦП, аппаратным умножителем 8х8.

Интересной особеннотью семейства МК является то, что система комманд всего семейства совместима при переносе программы со слабого на более мощный микроконтроллер [1- ст. 11].

1. Структура микроконтроллера ATMega128

В качестве ядра микроконтроллерной системы для измерения температуры в диапазоне 30-400С согласно техническому заданию был выбран AVR микроконтроллера типа ATMega128.

1.1 Назначение выводов

На рис.1.1 изображен корпус и приведено назначение выводов микроконтроллера. В скобках указана альтернативная функция вывода.

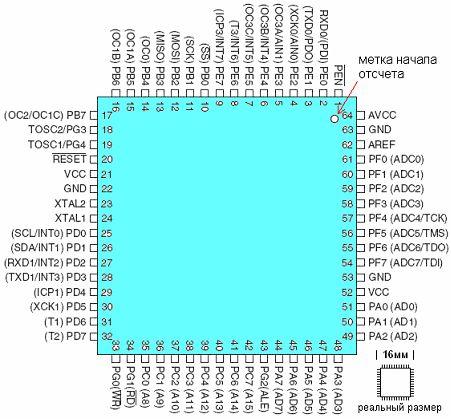


Рис.1.1. Вид корпуса и обозначение выводов микроконтроллера ATmega128.

Микроконтроллер ATmega128 включает следующие функциональные блоки: - 8-разрядное арифметическо-логическое устройство ( АЛУ );

- внутреннюю флэш-память программ объемом 128 Кбайт с возможностью внутрисистемного программирования через последовательный интерфейс;

- 32 регистра общего назначения;

- внутреннюю EEPROM память данных объемом 4 Кбайт;

- внутреннее ОЗУ данных объемом 4 Кбайт;

- 6 параллельных 8-разрядных портов;

- 4 программируемых таймера-счетчика;

- 10-разрядный 8-канальный АЦП и аналоговый компаратор;

- последовательные интерфейсы UART0, UART0, TWI и SPI;

- блоки прерывания и управления (включая сторожевой таймер).

Port A (PA7..PA). 8-разрядный двунаправленный порт. К выводам порта могут быть подключены встроенные нагрузочные резисторы (отдельно к каждому разряду). Выходные буферы обеспечивают ток 20 мА и способность прямо управлять светодиодным индикатором. При использовании выводов порта в качестве входов и установке внешнего сигнала в низкое состояние, ток будет вытекать только при подключенных встроенных нагрузочных резисторах. Порт А при наличии внешней памяти данных используется для организации мультиплексируемой шины адреса/данных.

Port B (PB7..PB0). 8-разрядный двунаправленный порт со встроенными нагрузочными резисторами. Выходные буферы обеспечивают ток 20 мА. При использовании выводов порта в качестве входов и установке внешнего сигнала в низкое состояние, ток будет вытекать только при подключенных встроенных нагрузочных резисторах. Порт В используется также при реализации специальных функций.

Port C (PC7..PC0). Порт С является 8-разрядным выходным портом. Выходные буферы обеспечивают ток 20 мА. Порт C при наличии внешней памяти данных используется для организации шины адреса.

Port D (PD7..PD0). 8-разрядный двунаправленный порт со встроенными нагрузочными резисторами. Выходные буферы обеспечивают ток 20 мА. При использовании выводов порта в качестве входов и установке внешнего сигнала в низкое состояние, ток будет вытекать только при подключенных встроенных нагрузочных резисторах. Порт D используется также при реализации специальных функций.

Port Е (PЕ7..PЕ0). 8-разрядный двунаправленный порт со встроенными нагрузочными резисторами. Выходные буферы обеспечивают ток 20 мА. При использовании выводов порта в качестве входов и установке внешнего сигнала в низкое состояние, вытекающий через них ток обеспечивается только при подключенных встроенных нагрузочных резисторах. Порт E используется также при реализации специальных функций.

Port F (PF7..PF0). 8-разрядный входной порт. Входы порта используются также как аналоговые входы аналого-цифрового преобразователя.

#RESET. Вход сброса. Для выполнения сброса необходимо удерживать низкий уровень на входе более 50 нс.

XTAL1, XTAL2. Вход и выход инвертирующего усилителя генератора тактовой частоты.

TOSC1, TOSC2. Вход и выход инвертирующего усилителя генератора таймера/счетчика.

#WR, #RD. Стробы записи и чтения внешней памяти данных.

ALE. Строб разрешения фиксации адреса внешней памяти. Строб ALE используется для фиксации младшего байта адреса с выводов AD0-AD7 в защелке адреса в течение первого цикла обращения. В течение второго цикла обращения выводы AD0-AD7 используются для передачи данных.

AVCC. Напряжение питания аналого-цифрового преобразователя. Вывод подсоединяется к VCC через низкочастотный фильтр.

AREF. Вход опорного напряжения для аналого-цифрового преобразователя. На этот вывод подается напряжение в диапазоне между AGND и AVCC.

AGND. Это вывод должен быть подсоединен к отдельной аналоговой земле, если она есть на плате. В ином случае вывод подсоединяется к общей земле.

#PEN. Вывод разрешения программирования через последовательный интерфейс. При удержании сигнала на этом выводе на низком уровне после включения питания, прибор переходит в режим программирования по последовательному каналу.

VСС, GND. Напряжение питания и земля [2-ст.7].

1.2 Организация памяти и портов ввода/вывода микроконтроллера

Микроконтроллеры AVR имеют раздельные пространства адресов памяти программ и данных (гарвардская архитектура). Организация памяти МК ATMega128 показана на рис. 1.2.

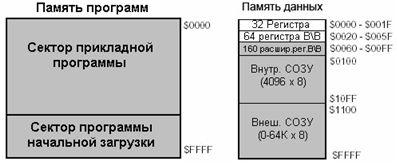


Рис.1.2. Организация памяти микроконтроллера ATmega128

Высокие характеристики семейства AVR обеспечиваются следующими особенностями архитектуры:

• В качестве памяти программ используется внутренняя флэш-память. Она организована в виде массива 16-разрядных ячеек и может загружаться программатором, либо через порт SPI;

• 16-разрядные память программ и шина команд вместе с одноуровневым конвейером позволяют выполнить большинство инструкций за один такт синхрогенератора (50 нс при частоте FOSC=20 МГц);

• память данных имеет 8-разрядную организацию. Младшие 32 адреса пространства занимают регистры общего назначения, далее следуют 64 адреса регистров ввода-вывода, затем внутреннее ОЗУ данных объемом до 4096 ячеек. Возможно применение внешнего ОЗУ данных объемом до 60 Кбайт;

• внутренняя энергонезависимая память типа EEPROM объемом до 4 Кбайт представляет собой самостоятельную матрицу, обращение к которой осуществляется через специальные регистры ввода-вывода.

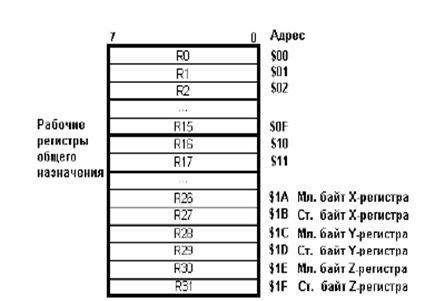


Рис.1.3. Регистры общего назначения микроконтроллера ATmega128

Как видно из рис. 1.2 и 1.3, 32 регистра общего назначения (РОН) включены в сквозное адресное пространство ОЗУ данных и занимают младшие адреса. Хотя физически регистры выделены из памяти данных, такая организация обеспечивает гибкость в работе. Регистры общего назначения прямо связаны с АЛУ. Каждый из регистров способен работать как аккумулятор. Большинство команд выполняются за один такт, при этом из регистров файла могут быть выбраны два операнда, выполнена операция и результат возвращен в регистровый файл. Старшие шесть регистров могут использоваться как три 16-разрядных регистра, и выполнять роль, например, указателей при косвенной адресации.

Следующие 64 адреса за регистрами общего назначения занимают регистры ввода-вывода (регистры управления/состояния и данных). В этой области сгруппированы все регистры данных, управления и статуса внутренних программируемых блоков ввода-вывода. При использовании команд IN и OUT используются адреса ввода-вывода с $00 по $3F. Но к регистрам ввода-вывода можно обращаться и как к ячейкам внутреннего ОЗУ. При этом к непосредственному адресу ввода-вывода прибавляется $20. Адрес регистра как ячейки ОЗУ приводится далее в круглых скобках. Регистры ввода-вывода с $00 ($20) по $1F ($3F) имеют программно доступные биты. Обращение к ним осуществляется командами SBI и CBI, а проверка состояния – командами SBIS и SBIC [2-ст.9].

2. Синтез структурной схемы

Структурная схема для цифрового термометра приведена на рисунке 2.1.



Рисунок 2.1 – структурная схема цифрового термометра

На рисунке 2.1 показано:

ЖКИ – жидко-кристаллический индикатор;

МК – микроконтроллер;

Д– цифровой датчик температуры;

RS232С – последовательный интерфейс.

Микроконтроллер выполняет две основные функции:

производит опрос датчика температуры и сохраняет в ОЗУ значения температуры, полученные от датчика в каждом цикле опроса температуры;

по требованию компьютера, микроконтроллер отсылает в компьютер значения температуры из ОЗУ от датчика температуры.

Измеренная температура не только сохраняется, но и выводится на жидкокристаллический экран (ЖКИ). Графические возможности экрана позволяют отображать не только цифровые значения температуры, но и отображать изменения температуры во времени в виде графиков. Также микроконтроллер может сохранять не одно значение температуры, а несколько (до 20 значений).

Если значение температуры выходит за диапазон 30-400С, то микроконтроллер формирует предупреждающий сигнал с помощью светодиодов.

Кнопка производит сброс, если измеренные значения температуры выходят за значения 30-400С.

При необходимости измеренные значения температуры можно ввести на экран компьютер с помощью последовательного интерфейса RS232С.

3. Синтез схемы электрической принципиальной цифрового термометра

В данном проекте разработан цифровой термометр с использованием микроконтроллера AVR ATMEGA128. Схема устройства представлена на чертеже ЛОЕТ.06-1.09Э3. Перечень элементов –ЛОЕТ.06-1.09 ПЭ3.

Описание схемы устройства

Прибор осуществляет измерения путём заряда конденсатора до уровня примерно равного VCC, последующего разряда его через опорный резистор с одновременным подсчётом внутренних тактов до того момента, пока на входе CIN не появится лог. «0». Далее конденсатор снова заряжается до значения, близкого к VCC и разряжается через термистор, при этом также подсчитываются тактовые импульсы. Неизвестное сопротивление резистора вычисляется как отношение числа тактов при разряде конденсатора термистором к числу тактов при разряде через опорный резистор и последующим домножением на известное значение сопротивления опорного резистора. Программа вычисляет сопротивле-ние термистора, переводит это значение в температуру, переводит её в градусы Цельсия и отображает значение на ЖКИ.

К выводам 50,51 МК подключен датчик температуры ТМР 35, представленная на рисунке 3.1



Рисунок 3.1- Микросхема TMP35

ТМР35 обеспечивает выходное напряжение, линейно изменяющееся пропорционально температуре по шкале Цельсия. TMP35 не требует внешней калибровки обеспечивая типичную точность +/-10C при +250C и +/-20C в диапазоне -400C к +1250C. Низкое выходное сопротивление TMP35 и линейность выводного напряжения и отсутствие необходимости точной калибровки упрощает его подключение к электрической схеме терморегулятора, Аналого-цифровым преобразователя. Устройства питается по двухпроводной линии от источника напряжением от 2.7 V к 5.5 V. Питающий ток имеет величину значительно ниже 50мка, обеспечивая очень низкий саморазогрев — меньше чем 0.10C в невозмущенной атмосфере. Кроме того, заложена функция отключения обеспечивающая снижение питающего тока до величины меньше чем 0.5мка. TMP35 работает при температуре от 100C до 1250C [3]. Резисторы R12 и R13 ограничивают ток, проходящий через датчик.

К выводам 14 и 15 (PB4, PB5) микроконтроллера с помощью ограничиваю-щих резисторов R2 ÷ R4 подсоединяется выводами CLK и DI ЖКИ, выполненный на микросхеме НТ1611. Через ограничивающие резисторы R6, R6 к выводу VDD подводится напряжение питания от источника.

Жидкокристаллический индикатор показан на рисунке 3.2

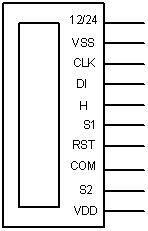


Рисунок 3.2- Микросхема НТ1611

Если измеренные значения температуры выходят за заданные пределы, то загораютя светодиоды VD1, VD2. R9,R10 ограничивают ток, проходящий через светодиоды.

К выводым 2, 3 (РЕ0,РЕ1) микроконтроллера присоединяется микросхема MAX232C, которая позволяет производить обмен информацией с компьютером через последовательный интерфейс RS232C. Для этого задействованы выводы T1IN, T1OUT, R1IN, R1OUT. В соответствии с описанием микросхемы к выводам C1+, C1- и C2+, C2- подключаются конденсаторы С3, С4 емкостью 0,1 мкФ. Такие же конденсаторы С1, С2 для создания необходимых уровней питания подключаются к ножке 2 (Vs+) и к шине питания; к ножке 6 (Vs-) и к земле [1]. Устройство можно подключить к компьютеру с помощью разъема XР1.

Условное графическое обозначение микросхемы MAX232C показан на рисунке 3.3.

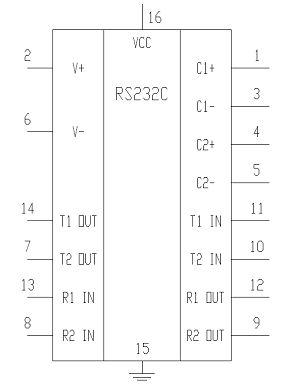


Рисунок 3.3 – УГО ADM202\_SO16

Микросхема ADM202\_SO16 позволяет осуществлять связь по интерфейсу RS232C одновременно с двумя устройствами.

Для формирования правильного импульса сброса в момент включения питания к выводу (RESET) микроконтроллера подключена RC-цепочка (R11,C9). Эта цепь используется для задержки запуска микроконтроллера при включении питания, что нужно для его правильного запуска, а также для ручного перезапуска микроконтроллера нажатием на кнопку SВ1 . Цепь сброса по включению питания обеспечивает запрет включения процессора до тех пор, пока напряжение питания не достигнет безопасного уровня. После того, как напряжение питания достигнет уровня включения, процессор не включается до тех пор, пока встроенный таймер не обработает несколько периодов сторожевого таймера. Внешний сброс обрабатывается по низкому уровню на выходе RESET. Вывод должен удерживаться в низком состоянии, по крайней мере, два периода тактовой частоты. После снятия сигнала 0 с вывода RESET через некоторое время микроконтороллер запускается. Кроме того, для информирования пользователя о рабочем режиме подключается светодиод VD3. Этот светодиод мигает зеленым цветом, когда производится чтение значений температуры из датчика. В остальное время светодиод не горит. Так как чтение значений температуры происходит непрерывно, кроме случаев прерывания вызванных для связи с компьютером, то светодиод мигает с периодом 0,8 мс. И перестает мигать в момент обмена информацией с компъютером [1].

К выводам 23 (XTA/L1) и 24 (XTA/L1) МК подсоединяется кварцевый резонатор ZQ1 с частотой 14 МГц. Для облегчения запуска тактового генератора необходимо подключение двух конденсаторов С7 и С8.

Устройство получает питание от аккумулятора напряжением 5 В через сглаживающий С-фильтр, состоящий из конденсаторов С5и С6 и стабилизатора напряжения. К вспомогательным устройствам относятся автоматический выключатель QF1 и индикатор включения – светодиод VD5. Резистор R7 ограничивает ток через светодиод. Стабилизатор напряжения включается между выходом выпрямителя и нагрузкой. Он выполнен на микросхеме DD3 типа КР142ЕН5А.

В качестве резисторов были выбраны резисторы МЛТ. Это металлопленочные резисторы, которые содержат резистивный элемент в виде очень тонкой металлической пленки, осажденной на основание из керамики, стекла, слоистого пластика, ситалла или другого изоляционного материала. Металлопленочные резисторы характеризуются высокой стабильностью параметров, слабой зависимостью сопротивления от частоты и напряжения и высокой надежностью. Недостатком некоторых металлопленочных резисторов является пониженная надежность при повышенной номинальной мощности, особенно при импульсной нагрузке. ТКС резисторов типов МЛТ не превышает 0,02% °С. Уровень шумов резисторов группы А не более 1 мкВ/В, группы Б – не более 5 мкВ/В.

Конденсаторы С6 - типа К53 - 4 – электролитический, оксиднополупроводниковый. Конденсаторы С1…С5, С9,С10 типа КМ-6– это керамические монолитные конденсаторы, широко применяется в разнообразных контурах и цепях радиоаппаратуры УКВ и КВ в качестве блокировочных, переходных и т.д.

Конденсаторы С7,C8 типа КЛС – конденсаторы керамческие литые секционные. Это конденсаторы постоянной емкости, общего назначения, которые предназначены для работы в цепях постоянного, переменного и импульсного тока.

Программирование МК осуществляется через разъем XP2.

4. Разработка алгоритма работы микроконтроллера

Программа работы микроконтроллера заключается в следующем:

при нажатии кнопки RESET (SB1) производится непрерывный (циклический) опрос датчика и сохранение полученных значений температуры в ОЗУ.

непрерывлый вывод полученных значений температуры на ЖКИ индикатор (цифровое отображение информации на экране)

формирование сигнала предупреждения с помощью блока светодиодов в случае выхода значения температуры за пределы 300С<Т<400С (согласно техническому заданию).

Алгоритм программы приведен на рисунке 4.1.

Первым действием в программе производятся начальные установки микроконтроллера. В них устанавливается указатель стека на последнюю ячейку ОЗУ, исходное состояние каналов связи с датчиками температуры и UART, скорость обмена по UART, разрешаются прерывания от таймера/счетчика0 и от UART, переписывается количество и индивидуальные адреса датчиков температуры из EEPROM в ОЗУ, в регистры записываются необходимые константы.

Когда начальные установки завершены, начинается часть программы, которая производит опрос датчика температуры. Она будет циклически повторятся, пока подводится питание к микроконтроллеру или пока не возникнет запрос на прерывание. Опрос датчика температуры начинается с сигнала сброса на линии (блок 2 рисунок 4.1) Затем следует команда игнорирования адреса датчика температуры SKIP ROM [CCh].

Команда начала измерения температуры CONVERT T [44h] (блок 4, рисунок 4.1) разрешает преобразование значений температуры в цифровой вид для датчика.

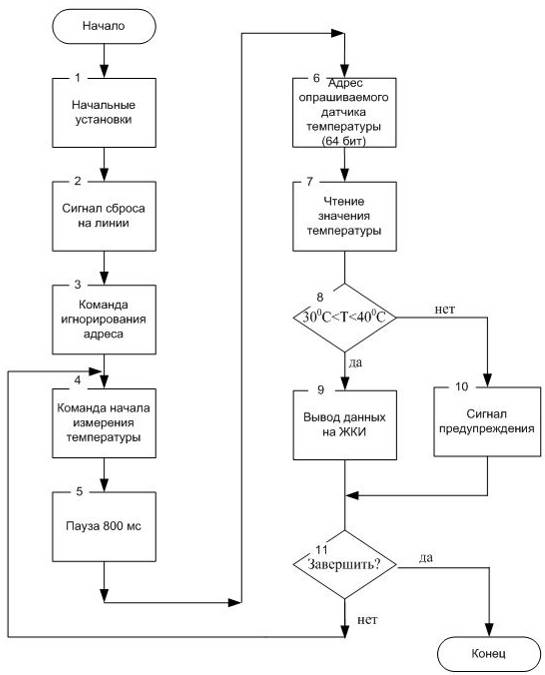


Рисунок 4.1 – Алгоритм работы микроконтроллера

Аналого-цифровое преобразование значений температуры занимает время от 750 мс до 800 мс. Поэтому, чтобы получить правильное значение температуры, необходимо выждать паузу 800 мс (блок 5, рисунок 4.1). Пауза выдерживается с помощью таймера/счетчика 0. Во время паузы можно совершать другие действия (например, произвести обмен данными с компьютером или вывести результаты на ЖКИ).

После паузы производится опрос датчика. Опрос датчика начинается с сигнала сброса на линии связи с датчиком (блок 6, рисунок 4.1). После сигнала сброса и ответного сигнала от датчика следует команда MATCH ROM [55h]. Эта команда сообщает датчику, что после неё на линии связи будет выставлен индивидуальный 64-х битовый адрес датчика. После того, как адрес выставлен на линии, датчик температуры сравнивает выставленный адрес со своим собственным адресом, и, после этого к работе с микроконтроллером датчик готов.

В блоке 7 производится чтение значения температуры и запись его в соответствующие ячейки ОЗУ.

В блоке 8 производится ветвление программы: если измеренное значение температуры не выходит за пределы 300С<Т<400С , то результат выводится на экран ЖКИ (блок 9). Если же измеренное значение температуры выходит за пределы диапазона, то загорятся световые индикаторы: VD1- если температура меньше 30 0С или VD2 –если больше 400С (блок 10).

Если необходимо продолжать измерять температуру (блок 11), то переходят к блоку 4, если нет, то тогда происходит завершение программы.

Программа работы микроконтроллера для измерения температуры приведена в приложении А.

Заключение

В данном курсовом проекте был разработан цифровой термометр, позволяющий снимать температуру датчиком температуры ТМР35, обрабатывать полученную информацию и выводить результат измерений в цифровом виде на экран ЖК индикатора. Также с помощью последовательного интерфейса возможна их передача на экран персонального компьютера для получения и сохранения неограниченного количество этих значений, проведения статистического анализа и слежение за изменением температуры во времени для разных этапов измерения.

В ходе проектирования были разработаны структурная и принципиальная электрическая схемы, алгоритм работ и текст программы для микроконтроллерной системы.

Таким образом, разработанный цифровой термометр является компактным, переносным устройством, дающим точную информацию о температуре, что позволяет расширить возможности измерений.

Список источников информации

1. Голубцов М.С., Кириченкова А.В. Микроконтроллеры AVR: от простого к сложному. Изд.2-е, испр. И доп. – М.: СОЛОН- Пресс, 2006. 304с.- (Серия «Биб-лиотека инженера»).

2. Китаев Ю.В. Основы программирования микроконтроллеров AT MEGA128 и 68HC908. Учебное пособие : СПб: СпбГУ ИТМО, 2007, 107с.

УДК 681.32

3. Low Voltage Temperature Sensors. TMP35/TMP36/TMP37

4. http://pdfserv.maxim-ic.com/arpdf/DS18S20.pdf. Датчики для измерения температуры.

5. Температурные измерения. Справочник./ Геращенко О.А. Гордов А.Н., Еремина А.К., и др.; отрв. Ред. Геращенко О.А.; АН УССР Ин-т проблем энергосбережения. – Киев: Наук. думка, 1989г. 704 с.

6. http://www.chip-dip.ru/product0/61922.aspx Кнопки тактовые.

Приложение А

Текст программы

; Системные определения для 320 STK/EVK

TOOL SET 0 ; 0 = STK/EVK ОЗУ

; 1 = Симулятор

; 2 = Встроенное ПЗУ

STACK EQU 003DEH ; Указатель стека

RAM\_ORIG EQU 00200H ; Начальный адрес памяти ОЗУ

ROM\_ORIG EQU 0C100H ; Начальный адрес памяти ПЗУ

IF TOOL = 0

I\_VECTORS EQU 003FFH ; Вектора прерываний в ОЗУ

MAIN EQU RAM\_ORIG+20H ; Стартовый адрес программы в ОЗУ

BTLOAD EQU 035H ; Прерывания раз в 0,5 сек.

ELSEIF TOOL = 1

I\_VECTORS EQU 0FFFFH ; Вектора прерываний в ПЗУ

MAIN EQU ROM\_ORIG ; Стартовый адрес программы в ПЗУ

BTLOAD EQU 011H ; Частые прерывания (не 1 сек)

ELSE

I\_VECTORS EQU 0FFFFH ; Вектора прерываний в ПЗУ

MAIN EQU ROM\_ORIG ; Стартовый адрес программы в ПЗУ

BTLOAD EQU 035H ; Прерывания раз в 0,5 сек.

.ENDIF

; Определения для АЦП на базе модуля TIMER PORT

TPCTL EQU 04BH ;Рег.управ.модулем TIMER PORT (04BH)

TPSSEL0 EQU 040H ;Источник тактирования:0=CMP,

;1=ACLK (Бит 6 в TPCTL)

ENB EQU 020 ;Контроль сигнала EN1 в TPCNT1

;1(+ENA=1)=CMP (Бит 5 в TPCTL)

ENA EQU 010H ;Контроль сигнала EN1 в TPCNT1

;1(+ENB=1)=CMP (Бит 4 в TPCTL)

EN1 EQU 008H ;Сигнал ENABLE в TPCNT1 только

;для чтения (Бит 3 в TPCTL)

RC2F EQU 004H ;Флаг перепол.TPCNT2(Бит2в TPCTL)

EN1FG EQU 001H ;Флаг EN1 (Бит 0 в TPCTL)

TPIE EQU 004H ;Разреш.прерыв.от TIMER(Бит3вIE2)

TPCNT1 EQU 04CH ;Младший байт счётчика

TPCNT2 EQU 04DH ;Старший байт счётчика

TP EQU 04EH ;Регистр данных TP(0–5=Выход TP,

;6=CPON,7=B16=2x8 Бит или1x16Бит

B16 EQU 080H ;Раздел.тайм-ы(0)или один16-бит(1)

CPO EQU 040H ;Компаратор выкл (0) / вкл (1)

TPDMAX EQU 002H ;бит вых.TPD.MAX(2=Бит1=TPD.1)

TPE EQU 04FH ;Рег.разреш.выходов 0–5=биты разреш

.TPD (6–7=ист.тактир.TPCNT2)

MSTACK EQU 03D2H ;1-е слово стека результата

PRESET EQU 0E8H ;Предустановка TPCNT2 для заряда С

;счёт останавливается при

;переполнении TPCNT2, значение

;рассчитано на постоянную врем.6RC

; Определения управляющих регистров

IE1 EQU 0H ;Регистр разрешения прерываний 1

IE2 EQU 01H ;Регистр разрешения прерываний 2

P01IE EQU 08H ;Разрешение прерываний отP0.1в IE1

BTIE EQU 080H ;Разреш.прерыв.от BASIC TIMER в IE2

IFG1 EQU 02H ;Регистр флагов прерываний 1

IFG2 EQU 03H ;Регистр флагов прерываний 2

LCDCTL EQU 030H ;Регистр управления ЖКИ

LCDM1 EQU 031H ;Первая ячейка памяти ЖКИ

BTCTL EQU 040H ;Регистр управления BASIC TIMER

BTCNT1 EQU 0046H ;Счётчик 1 BASIC TIMER

BTCNT2 EQU 0047H ;Счётчик 2 BASIC TIMER

WDTCTL EQU 0120H ;Регистр управления WATCHDOG

WDTHOLD EQU 080H ;Маска останова WATCHDOG

WDT\_KEY EQU 05A00H ;Ключ доступа к WATCHDOG

WDT\_STOP EQU 05A80H ;WATCHDOG маска + ключ

GIE SET 8H ;Общее разрешение прерываний

CPUOFF SET 10H ;Бит выключения CPU

OSCOFF SET 20H ;Бит выключения генератора

SCG0 SET 40H ;Контр.сист.тактирования,бит0

SCG1 SET 80H ;Контр.сист.тактирования,бит1

LPM0 SET CPUOFF ;Биты установки режима LPM0

LPM1 SET SCG0+CPUOFF ;Биты установки режима LPM1

LPM2 SET SCG1+CPUOFF ;Биты установки режима LPM2

LPM3 SET SCG1+SCG0+CPUOFF ;Биты установки режима LPM3

LPM4 SET OSCOFF+CPUOFF ;Биты установки режима LPM4

; Служебные регистры для вычисления сопротивления датчика

MLTPLR\_HW EQU R5

TEN\_K EQU R6

BITTEST EQU R7

MRESLT\_HW EQU R8

MRESLT\_LW EQU R9

LPCNTR EQU R10

RESULT EQU R11

; Начало программы

SECT ”MAIN”,MAIN

RESET MOV #STACK,SP ;Инициализация указателя стека

; Конфигурация периферии

SETUP

SETUPINT MOV.B #P01IE,&IE1 ;Разреш.P0.1/UART для S232

MOV.B #BTIE+TPIE,&IE2 ;Разреш.прерыв.от B.TIMER,&TMR.PORT

CLR.B &IFG1 ; Очистить все флаги прерываний

CLR.B &IFG2

EINT ;Разрешить прерывания

SETUPWDT MOV #WDT\_STOP,&WDTCTL ; Остановить WATCHDOG TIMER

SETUPLCD MOV.B #0FFH,&LCDCTL ; ЖКИ STK, все сегменты, 4MUX

SETUPBT MOV.B #BTLOAD,&BTCTL ; Загруз.част.прерыв в BASIC TIMER

CLR.B &BTCNT1 ; Очистить счётчик BT 1

CLR.B &BTCNT2 ; Очистить счётчик BT 2

CLEARLCD MOV #15,R6 ; Очистить 15 ячеек памяти ЖКИ

CLEAR1 MOV.B #0,LCDM1–1(R6) ; записав туда «0»

DEC R6 ; Вся памяти ЖКИ очищена?

JNZ CLEAR1 ; нет, чистим дальше

; Начало основной программы

BEGIN BIS #LPM3,SR ; Установить SR-биты для LPM3

;Подпрограмма измерения с запрещёнными прерываниями. TP.2–.5 не

;используются поэтому переписаны. Используются только TPD.0 & 1.

;Начальная инициализация: Указатель стека = 0, Начало с TPD.1

;16–битный таймер, тактирование от MCLK, CIN разрешает счёт MEASURE PUSH.B #TPDMAX ;Сохранить в стеке для послед.использ.

CLR R8 ;Указатель для стека результата

MEASLOP MOV.B #(TPSSEL0\*3)+ENA,&TPCTL

;TPCNT1 CLK=MCLK, EN1=1

;Конденсатор С заряжается на протяжении t>5нс.Используются выходы N–1

MOV.B #B16+TPDMAX–1,&TPD;один16-бит.счёт,выбор вых. заряда

MOV.B #TPDMAX–1,&TPE ; Разрешить выходы заряда

MOV.B #PRESET,&TPCNT2 ; Загрузить время заряда

BIS #CPUOFF,SR ; Режим понижен.потребления

MOV.B @SP,&TPE ;Разрешить только текущий датчик

CLR.B &TPCNT2

; Запрет.все прерыв.для обеспеч.непрерыв.работы тайм.и разряд конден.

DINT ; Запрет прерываний

CLR.B &TPCNT1 ; Очистить младший байт таймера

BIC.B @SP,&TPD ; Переключить все датчики в «0»

MOV.B #(TPSSEL0\*3)+ENA+ENB,&TPCTL ;TPCNT1 CLK=MCLK,

; разрешить вход CIN

EINT ; Разрешить прерывания, общий старт

BIS #CPUOFF,SR ; Режим пониженного потребления

; EN=0:Завершение преобразования: 2X8 бит результата в MSTACK

; Сохранить результат в стеке

MOV.B &TPCNT2,MSTACK+1(R8)

; Сохранить старший байт результата

L$301 INCD R8 ; Инкремент адреса

RRA.B @SP ; Следующий выход TPD.X

JNC MEASLOP ; Если C=1 – завершение

INCD SP ; Убрать старший TPD из стека

; Вычисление сопротивления датчика

;Подпрограмма беззнакового умножения:MSTACK X TEN\_K MRESLT\_HW/MRESLT\_LW

;Использованы рег.MSTACK,TEN\_K,MLTPLR\_HW,MRESLT\_LW, MRESLT\_HW, BITTEST

;Подпрограмма беззнакового умножения с накоплением

;(MSTACK X TEN\_K) + MRESLT\_HW|MRESLT\_LW –> MRESLT\_HW|MRESLT\_LW

CALC\_RES

MOV #10000,TEN\_K ;Загрузить 10,000 десятичное в TEN\_K

MPYU CLR MRESLT\_LW ; 0 ? LSBS результата

CLR MRESLT\_HW ; 0 ? MSBS результата

MACU CLR MLTPLR\_HW ; 0 ? MSBS множителя

MOV #1,BITTEST ; Регистр проверки бит

L$002 BIT BITTEST,MSTACK ; Проверить текущий бит

JZ L$01 ; Если «0» ничего не делать

ADD TEN\_K,MRESLT\_LW ;Если «1»добавить множитель к резул

ADDC MLTPLR\_HW,MRESLT\_HW

L$01 RLA TEN\_K ; Множитель X 2

RLC MLTPLR\_HW

RLA BITTEST ; Проверить следующий бит

JNC L$002 ;Если бит в CARRY: завершить

; Подпрограмма беззнакового деления 32–бита на 16-бит

; Использованы регистры (MSTACK+2),MRESLT\_LW,RESULT, LPCNTR, MRESLT\_HW

; MRESLT\_HW MRESLT\_LW / (MSTACK+2) RESULT Остаток в MRESLT\_HW

; По выходу: CARRY = 0: OK CARRY = 1: Частное > 16 Бит

DIVIDE CLR RESULT ; Очистить RESULT

MOV #17,LPCNTR ; Инициализация счётчика

DIV1 CMP MSTACK+2,MRESLT\_HW

JLO DIV2

SUB MSTACK+2,MRESLT\_HW

DIV2 RLC RESULT

JC RES\_2\_F ;Ошибка: RESULT > 16 Бит

DEC LPCNTR ; Декремент счётчика

JZ DIV3 ; 0: выход без ошибки

RLA MRESLT\_LW

RLC MRESLT\_HW

JNC DIV1

SUB MSTACK+2,MRESLT\_HW

SETC

JMP DIV2

DIV3 CLRC ;Ошибки нет, C = 0

; Перевод сопротивления датчика в градусы Цельсия для отображения RES\_2\_F

CLR R12 ;Указат.на значение в табл.R

MOV #064H,R13 ;Поместить мин.темп.-1 в индик

JMP FIRST\_CMP ;При первом сравн.1 не добав.

CHECK\_R INCD R12 ;INC указат.на знач.в таб.cопрот.

DADD #1,R13 ;Десятичный инкремент счётчика

FIRST\_CMP CMP RESIS\_TAB(R12),RESULT

;Сравнить табличное значение с вычисленным сопротивлением

JNC CHECK\_R ;Переход,если R датч.<табличного

;по адресу указателя

;Отобразить ”С” и знак «°» на ЖКИ

DISPLAY MOV.B #A+E+F+D,LCDM1+1 ;”С” ? дисплейная память ЖК

MOV.B #A+B+F+D,LCDM1+2 ;” °” (знак градуса)

; Отобразить значение, хранящееся в R13 в формате BCD на ЖКИ

MOV R13,R12 ;Копир.число в виде BCD в R12

MOV #LCDM1+4,R14 ;Младшая цифра в памяти ЖКИ; R14

BIC #0FFF0H,R13 ;Погасить всё,кроме младшей цифры

MOV.B LCD\_TAB(R13),0(R14) ; Отпр.мл.цифру в ЖКИ

MOV R12,R13 ; Восстановить значение в R13

RRA R13 ; 4 сдвига

RRA R13

RRA R13

RRA R13

BIC #0FFF0H,R13 ; Погасить всё, кроме млад.цифры

MOV.B LCD\_TAB(R13),1(R14) ;Отправить старшую цифру в ЖКИ

JMP BEGIN ; Переход к началу программы

; ЖКИ модуля STK

LCD\_TYPE

A EQU01H

B EQU02H

C EQU10H

D EQU04H

E EQU80H

F EQU20H

G EQU08H

H EQU40H

LCD\_TAB BYTE A+B+C+D+E+F ; Отображает”0”

.BYTE B+C ; Отображает”1”

.BYTE A+B+D+E+G ; Отображает”2”

.BYTE A+B+C+D+G ; Отображает”3”

.BYTE B+C+F+G ; Отображает”4”

.BYTE A+C+D+F+G ; Отображает”5”

.BYTE A+C+D+E+F+G ; Отображает”6”

.BYTE A+B+C ; Отображает”7”

.BYTE A+B+C+D+E+F+G ; Отображает”8”

.BYTE A+B+C+D+F+G ; Отображает”9”

; Табл.сопротив.30–40 °C. Значения = KОмыX1000–в 3 децимальных цифры

.EVEN ; Выравнивание по чётному адресу

RESIS\_TAB

.WORD 12953 ;300C

.WORD 12666

.WORD 12378

.WORD 12090

.WORD 11858

.WORD 11626

.WORD 11393

.WORD 11161

.WORD 10929

.WORD 10353 ;400C

; Вектора прерываний

.EVEN ; Выравнивание по чётному адресу

.SECT ”INT\_VECT”,I\_VECTORS–31

.WORD RESET ; PORT0, Биты 2 .. 7

.WORD BTINT ; BASIC TIMER

.WORD RESET ; Нет источника

.WORD RESET ; Нет источника

.WORD RESET ; Нет источника

.WORD TPINT ; TIMER PORT

.WORD RESET ; Нет источника

.WORD RESET ; Нет источника

.WORD RESET ; Нет источника

.WORD RESET ; Нет источника

.WORD RESET ; WATCHDOG/TIMER, Режим таймера

.WORD RESET ; Нет источника

.WORD RESET ; UART

.WORD RESET ; P0.0

.WORD RESET ; NMI, Сбой генератора

.WORD RESET ; POR,Внеш. Reset, WATCHDOG

.END