*Государственное образовательное учреждение высшего образования*

*«Московский государственный технический университет*

*им. Н.Э. Баумана (национальный исследовательский университет)»*

*(МГТУ им. Н.Э. Баумана)*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*ФАКУЛЬТЕТ «СПЕЦИАЛЬНОЕ МАШИНОСТРОЕНИЕ»*

*КАФЕДРА «ПОДВОДНЫЕ АППАРАТЫ И РОБОТЫ»*

НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА СТУДЕНТА

СИСТЕМЫ ТЕХНИЧЕСКОГО ЗРЕНИЯ НЕОБИТАЕМОГО ПОДВОДНОГО АППАТАРА

Руководитель НИРС **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** (Плясунов В.М.)

(подпись, дата)

Исполнитель НИРС,

студент группы СМ11-71 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** (Каменев Я.М.)

(подпись, дата)

Москва, 2017

Оглавление

[Введение 2](#_Toc502309917)

[1. Описание технического задания 3](#_Toc502309918)

[2. Описание и обоснование схемы электрической принципиальной 4](#_Toc502309919)

[2.1 Выбор микроконтроллера 4](#_Toc502309920)

[2.2 Выбор драйвера двигателей постоянного тока 5](#_Toc502309921)

[2.3 Выбор источника тактового сигнала 6](#_Toc502309922)

[2.4 Выбор приемника/передатчика RS-485 7](#_Toc502309923)

[2.5 Выбор преобразователей напряжения 8](#_Toc502309924)

[2.7 Выбор разъемов 9](#_Toc502309925)

[3. Описание и обоснование конструкции 10](#_Toc502309926)

[4. Разработка программного обеспечения модуля управления ОТ 13](#_Toc502309927)

[4.1 Протокол обмена 13](#_Toc502309928)

[4.2 Основные блоки программного обеспечения 14](#_Toc502309929)

[4.2.1 Инициализация 14](#_Toc502309930)

[4.2.2 Основной цикл программы 15](#_Toc502309931)

[4.2.3 Подпрограмма прерывания окончания приема по USART 16](#_Toc502309932)

[4.2.4 Подпрограмма прерывания по опустошению UDR0 16](#_Toc502309933)

[4.2.5 Подпрограмма прерывания по завершению приема 16](#_Toc502309934)

[4.2.6 Подпрограмма прерывания по совпадению канала A T1 16](#_Toc502309935)

[4.2.7 Подпрограмма прерывания по завершению измерения АЦП 17](#_Toc502309936)

[5. Листинг программы 17](#_Toc502309937)

# Введение

Целью данной работы является проектирование модуля управления объективом-трансфокатором камеры подводного аппарата. Объектив-трансфокатор может менять фокусное расстояние и увеличение. Поскольку данное устройство позиционируется как самостоятельное, оно должно подключаться к управляющему устройству с помощью интерфейса RS-485.

В данной работе были разработаны схема электрическая принципиальная, конструкция печатной платы, сборочный чертёж модуля и программное обеспечение для микроконтроллера, используемого в данной схеме.

# Описание технического задания

Объектив-трансфокатор (ОТ) представляет собой объектив, который может изменять свое фокусное расстояние и увеличение. Фокус и увеличение изменяются при помощи двух двигателей постоянного тока (ДПТ), которые вращают линзы, установленные в передаче винт-гайка. Для замыкания обратной связи в контуре управления фокусом располагается потенциометр. Связь с управляющим устройством осуществляется по интерфейсу RS-485.

Устройство использует адресный протокол, структура которого:

Принимаемая посылка: 3 байта

1. Адресный байт
2. Байт управления драйвером ДПТ (младшие 4 бита)
3. Контрольная сумма

Отправляемая посылка: 5 байт

1. Адресный байт
2. Байт состояния выходов для управления драйвером ДПТ
3. Старшие 2 бита 10-битного АЦП
4. Младшие 8 битов 10-битного АЦП
5. Контрольная сумма

После получения посылки устройство отрабатывает полученную команду и передает управляющему устройству состояние выводов МК, управляющих драйвером ДПТ и результат считывания значения напряжения на потенциометре.

# Описание и обоснование схемы электрической принципиальной

В соответствии с требованиями технического задания, основным элементом конструкции является 8-разрядный микроконтроллер. Управление ДПТ должно осуществляться с помощью драйвера, которые, получив сигнал с МК, содержащий в себе направление вращения ДПТ, пропускает ток через обмотки двигателя в нужном направлении. На потенциометр подается опорное напряжение, формируемое на плате, оно же подается на вход опорного напряжения АЦП.

# Выбор микроконтроллера

В качестве главного исполнительного устройства обработки сигналов выбран восьмиразрядный микроконтроллер ATmega164A фирмы Atmel (позиционое обозначение DD3). Микроконтроллер содержит в одной микросхеме как вычислительное устройство (процессор), так и ряд периферийных устройств, таких как порты ввода/вывода, таймеры/счётчики и последовательные интерфейсы USART, аналого-цифровой преобразователь. Данный тип микроконтроллеров сочетает RISC и гарвардскую архитектуру, что позволяет реализовать высокую производительность обработки данных, которая достигает 1 MIPS. Ядро AVR сочетает в себе богатый набор инструкций и 32 рабочих регистра общего назначения. Каждый регистр общего назначения напрямую соединён с арифметико-логическим устройством, что позволяет обеспечить одновременный доступ к двум РОНам за один такт процессора.

Для данной схемы выбран 44-выводной TQFP корпус, изображенный на рис. 1. Данный микроконтроллер имеет следующие характеристики:

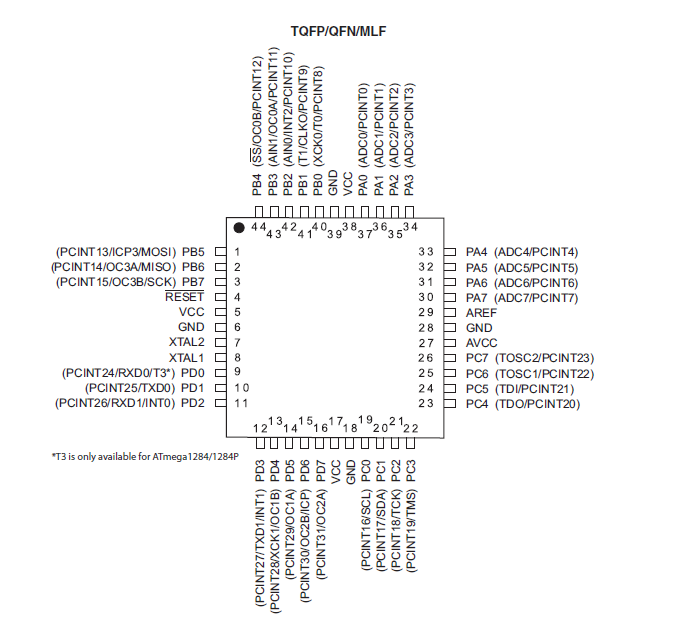


Рис. 1 – корпус TQFP-44 микроконтроллера ATmega164A

# Выбор драйвера двигателей постоянного тока

В качестве драйвера двигателей постоянного тока выбран драйвер LB1836M (позиционное обозначение DD1). Драйвер имеет два канала и позволяет управлять двумя двигателями. Максимальный выходной ток при напряжении 10,5 В составляет 1 А. Драйвер имеет по два управляющих вывода на каждый канал. Драйвер представляет из собой два H моста и имеет по два вывода для каждого канала, которые подключается к полюсам ДПТ. Таблица состояний драйвера в зависимости от уровней на управляющих пинах представлена на рис. 2.

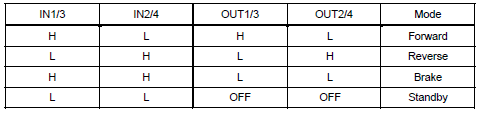


Рис. 2 – таблица режимов драйвера

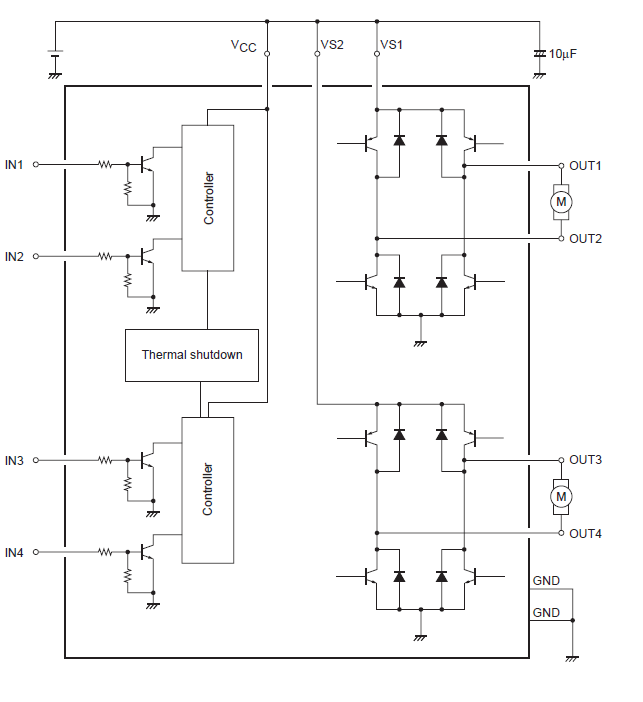


Рис. 3 – блок-схема драйвера

# Выбор источника тактового сигнала

В качестве источника тактового сигнала в данной схеме выбран кварцевый резонатор HC-49/SM (позиционное обозначение ZQ1) с рабочей частотой 8 МГц.

Кварцевые резонаторы являются пассивными компонентами радиоэлектронной аппаратуры и предназначены для использования в аналогово-цифровых цепях для стабилизации и выделения электрических колебаний определенной частоты или полосы частот. Принцип работы этого элемента следующий - в широкой полосе частот сопротивление прибора имеет емкостной характер и только на некоторых рабочих частотах имеет резко выраженный резонанс (уменьшение сопротивления).



Рис. 4 – Внешний вид кварцевого резонатора HC-49/SM

Кварцевый резонатор имеет лучшие характеристики, чем другие приборы для стабилизации частоты (колебательные контуры, пьезокерамические резонаторы): такие как стабильность по частоте (уход частоты) и температуре (изменение частоты резонанса в зависимости от температуры окружающей среды).

Избирательный, ярко выраженный резонансный характер сопротивления этих компонентов определяет основные области применения кварцевых резонаторов - высокостабильные генераторы тактовых сигналов и опорных частот, цепи частотной селекции, синтезаторы частоты.

Выводы кварцевого резонатора подключаются к выводам XTAL1 и XTAL2 микроконтроллера. Также к выводам необходимо подключить два керамических конденсатора с номиналом, указанным в документации на кварцевый резонатор. Для данной схемы номинал равен 22 пФ (позиционные обозначения C17, C18).

# Выбор приемника/передатчика RS-485

Для реализации интерфейса RS485 в данном модуле используется микросхема ADM2582 (позиционное обозначение DD2). Структурная схема данного устройства представлена на рис. 5.

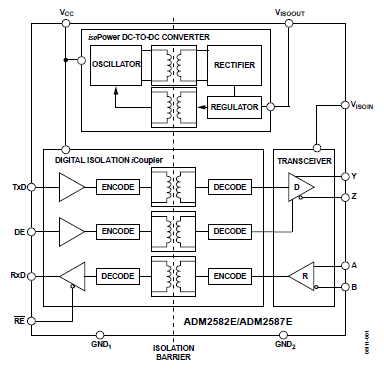


Рис. 5 – Структурная схема передатчика RS-485

Данная микросхема имеет внутри устройство приёмника и устройство передатчика. Данные передаются по двум проводникам, которые образуют витую пару, что позволяет обеспечить высокую помехозащищённость и увеличить длину линии связи.

Микросхема имеет в своем составе гальванические развязки линий связи и питания, что позволяет уменьшить количество DC-DC преобразователей напряжения, размещаемых на плате.

Передатчик запитан напряжением 5В и может передавать данные со скоростью до 16 Мб/сек.

# Выбор преобразователей напряжения

По условию технического задания плата питается напряжением 27 вольт. Так как номинальное напряжение микросхем составляет 5 В, необходимо использовать в микросхеме DC-DC преобразователи.

Микроконтроллер потребляет 0,4 мА, приемник-передатчик до 15 мА, источник опорного напряжения поддерживает выход до 5 мА. Драйвер ДПТ может поддерживать выход до 500 мА, но для маломощных ДПТ достаточно и 200 мА.

В данном проекте выбраны два преобразователя фирмы Traco Power TMR 1-2411SM (позиционные изображения DA2, DA3). Данный тип преобразователей рассчитан на входное напряжение от 18 до 36 В, и выдает выходное напряжение 5В. Максимальный ток выхода составляет 200 мА.

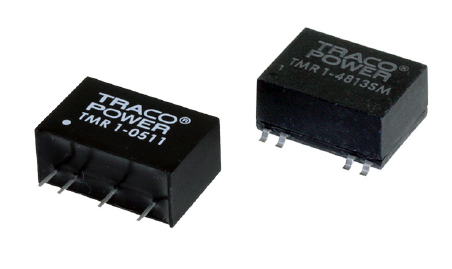


Рис. 6 – Внешний вид преобразователей

Для данной схемы выбран SMD корпус с 8-ю выводами. Микросхема работает в диапазоне температур от -40 ° до + 85 ° C.

В качестве источника опорного напряжения был выбран источник AD1584 (позиционное обозначение DA1), который преобразовывает 5В в 4,096В.

# 2.7 Выбор разъемов

В данном проекте используются несколько видов разъёмов.

Разъем CWF-2 (позиционное обозначение XP4) служит для подключения к плате источника питания на 27В. Рабочее напряжение разъема равно 250 вольт, максимальный ток 3 ампера, шаг контактов равен 2,54 мм. Разъем и вилка имеют конструктивный элемент, который не позволяет перепутать полюса подключаемого устройства. Для данного модуля был выбран прямой разъем.

Для подключения двигателей и потенциометра и RS-485 выбраны вилки PLS7 и PLS3. Контакты расположены в один ряд с шагом 2,54 мм.

Для подключения JTAG программатора выбрана вилка PLD10. Контакты расположены в два ряда с шагом 2,54 мм. Для отладки и прошивки микроконтроллера используется программатор JTAG ICE.



Рис. 7 – Внешний вид программатора JTAG ICE.

# Описание и обоснование конструкции

Для удовлетворения требованиям по габаритам, весу, устойчивости к механическим воздействиям (виброустойчивость, удароустойчивость) и воздействиям от окружающей среды (температуре, влажности) модуль управления ОТ будет выполнять на печатной плате.

Прежде чем начать трассировку, необходимо задать контур будущей печатной платы и создать крепежные отверстия. Выбрана плата прямоугольной формы с размерами 100х60 мм. Расположение крепежных отверстий на плате, во-первых, задается конструкцией блока, в который она будет входить, а во-вторых, массогабаритными характеристиками самой платы. Для небольших и легких плат достаточно четырех отверстий по углам платы, выбираем диаметры 3,3 мм и располагаем их на расстоянии 5 мм от краев платы.

Для трассировки печатных плат воспользуемся модулем PCB программного комплекса Altium Designer. Он служит для размещения посадочных мест компонентов на печатной плате и позволяет осуществлять трассировку печатных проводников, соединяющих выводы компонентов между собой в соответствии с электрической принципиальной схемой.

Следующий этап работы связан с рациональным размещением всех элементов устройства на поле печатной платы. Среди критериев рационального размещения элементов можно выделить:

* Группировка отдельно аналоговых элементов от цифровых;
* Минимизация длины печатных проводников;
* Разбиение схемы на отдельные функциональные блоки, при этом элементы, входящие в блоки, на плате располагаются рядом друг с другом;
* Учет электрофизических особенностей электронных устройств (сюда можно включить расположение фильтрующих конденсаторов и индуктивностей и т.д.);

Этот перечень критериев для каждого конкретного устройства может заметно меняться, дополняясь новыми требованиями.

Нередко оптимальное расположение элементов достигается в процессе последовательного многоэтапного совершенствования, перемещения или изменения расположения как отдельных элементов, так и целых групп, а также выполнения пробной трассировки всех или части проводников.

Обычно компоновка начинается с анализа электрической схемы с целью деления ее на отдельные блоки. Смысл такого деления заключается в том, что подавляющее большинство соединений должно быть между элементами, находящимися внутри блока. Поэтому такой блок можно представить в виде единого «большого элемента», который имеет соединения (небольшое количество) с другими блоками.

Переходные отверстия предназначены для обеспечения электрического соединения участков цепи, расположенных на разных слоях платы.

Конструктивно-технологические параметры переходных отверстий зависят от многих факторов: толщины платы, плотности монтажа, токовой нагрузки цепи и т.д.

Современное производство позволяет выполнять переходные отверстия диаметром 0,3 мм, но это повысит цену готового устройства. Наиболее распространенное переходное отверстие имеет диаметр 0,6 мм и поясок металлизации диаметром 1,2 мм.

Физически электрическая цепь на печатной плате является печатным проводником (дорожкой). Процесс трассировки печатной платы заключается в разводке по поверхности платы всех электрических цепей, которые на данном этапе представлены в виде соединений.

К конструктивно-технологическим параметрам печатных проводников относятся их ширина и поперечное сечение.

Поперечное сечение полностью зависит от технологии изготовления печатной платы. Ширина же зависит от ряда факторов: технологии изготовления, плотности монтажа, токовой нагрузки цепи и т.д.

Что касается токовой нагрузки, то на практике существует правило: проводник шириной 1 мм обеспечивает протекание тока 1 А.

Современное производство позволяет выполнять печатные проводники шириной 0,1 мм. Как правило, такая ширина необходима в случае, когда в схеме используются элементы с большим количеством выводов в корпусах с шариковыми выводами. В нашем случае компоненты такого типа не используются, поэтому выбираем ширину сигнальных проводников 0,3 мм, для проводников питания 5В 0,5 мм, 27В 1 мм.

Очень часто свободное пространство на печатной плате заполняется сплошным слоем металлизации (полигонами). Обычно этот слой подключатся к цепи «Общий» схемы. Сплошной слой металлизации обеспечивает качественную топологию общей цепи, а также играет роль экрана для сигнальных цепей. Величина зазора между полигоном и любым другим элементом топологии определяется в первую очередь разностью потенциалов между полигоном и прилегающими к нему элементами топологии. Здесь можно руководствоваться правилом: для напряжений до 50 В зазор должен быть не менее 0,5 мм.

Данная плата выполнена из двух слоев. На нижнем слое находятся полигоны, подключенные к одним из трех выводов «общий» и соответственно нижний слой, разделяется на три зоны (по количеству «общих» проводников в схеме). На верхнем слое располагается один полигон, подключенный к +5В, что позволяет развести питания на плате по схеме «звезда».

# Разработка программного обеспечения модуля управления ОТ

Программное обеспечение разрабатывается в специальной среде AVR Studio версии 4.1 на языке программирования AVR Assembler. Программное обеспечение необходимо для микроконтроллера, чтобы он успешно реализовал свои функции. Основные блоки программного обеспечения показаны на структурной схеме программы.

# Протокол обмена

В подавляющем большинстве протоколов обмена между устройствами применяются контрольные суммы для контроля сохранности данных для передачи. Так же протокол, использованный в этом проекте – адресный, что означает, что первый байт сообщения будет содержать в себе адрес устройства. В случае если адрес устройства не совпадает с адресом в сообщении – сообщение игнорируется. Члены сети данного протокола имеют разный вес, в данном случае master (ведущий) и slave (ведомый). Ведущий инициирует обмен и решает, кто может занимать шину в текущий момент времени. В нашем случае наша плата – ведомая.

Таким образом сообщение выглядит так:

Где – первые байты сообщения, полезная информация;

A - адрес, C – контрольная сумма (по одному байту)

Для вычисления контрольной суммы отрицание выполняется оператором COM языка Ассемблер микроконтроллера AVR. Суммирование выполняется по всем байтам полезной информации без учёта переполнения командой ADD.

Ведущий контроллер передает 3 байта следующей информации:

* 0-й байт – адресный байт;
* 1-й байт – байт команды;
* 2-й байт – байт контрольной суммы.

Младшие 4 бита байта команды соответствуют режимам работы драйвера, представленным выше.

Ответ (квитанция) контроллера ОТ ведущему состоит из четырех байт:

* 0-й байт – адресный байт;
* 1-й байт – байт состояния выводов управления драйвером;
* 2-й байт – старшие 2 бита результата работы АЦП;
* 3-й байт – младшие 8 бит результата работы АЦП;
* 4-й байт – байт контрольной суммы.

# Основные блоки программного обеспечения

## Инициализация

Самым первым этапом программного обеспечения является инициализация микроконтроллера. На данном этапе инициализируются порты ввода/вывода, последовательные приёмники/передатчики USART, таймеры/счётчики и необходимые прерывания.

Вывод PA0 является нулевым каналом аналого-цифрового преобразователя. Конфигурация порта выглядит следующим образом: 00000001.

Четыре вывода порта B служат для управления драйвером ДПТ. Конфигурация порта выглядит следующим образом: 11110000.

Четыре вывода порта C используются для отладки по интерфейсу JTAG. Конфигурация порта выглядит так: 00111100

Три вывода порта D используются контроллером USART и еще один нужен для обозначения направления передачи данных (flow control). Конфигурация порта выглядит следующим образом: 00000110.

16-битный таймер-счетчик T1 необходим для формирования интервала 6 мс, что соответствует времени таймаута по приему. Для реализации этой задачи инициализируется прерывание таймера по совпадению канала A, делитель частоты отключен. Верхний предел счета устанавливается в 0xBB80.

USART0 инициализируется с установкой режима межпроцессорного обмена (MPCM0 = 1), в асинхронном режиме и с включенным битом четности. Частота обмена данными равна 19200 бод.

Так же происходит инициализация стэка (указатель стэка устанавливается в конец ОЗУ), и очистка буферов приема и отправления, находящихся в ОЗУ МК, так же обнуление используемых РОН’ов.

## Основной цикл программы

В основном цикле программы проверяются флаги, устанавливаемые в прерываниях в таком порядке:

Флаг приема посылки управления FLAG\_RECEIVE

Окончание работы АЦП FLAG\_ADC\_READY

Между проверками установки флагов выполняются промежуточные действия:

Вывод управляющих сигналов на выводы МК

Формирование отправляемой посылки

После успешной проверки установки последнего флага цикл переходит к первой проверке.

## Подпрограмма прерывания окончания приема по USART

Данная подпрограмма выполняет подсчет и сохранение принятых байт в буфер приема, находящийся в ОЗУ МК. Так же выполняется проверка наличия ошибок приема и проверка адреса. В случае отсутствия ошибок и совпадения адреса бит MPCM устанавливается в 0, что означает разрешения приема не адресных байт. Так же стартует таймер T1, который отсчитывает время таймаута приема. При поступлении последнего байта (контрольной суммы) проверяется ее правильность.

## Подпрограмма прерывания по опустошению UDR0

Данная подпрограмма выполняет разрешение прерывания по завершению передачи и загрузку адреса начала буфера передачи в адресный регистр Y.

## Подпрограмма прерывания по завершению приема

В этой подпрограмме выполняется подсчет отправленных байт и их загрузка из буфера отправления. При завершении отправки бит MPCM устанавливается в 1, так же разрешается прием по USART.

## Подпрограмма прерывания по совпадению канала A T1

В данной подпрограмме происходит запрет прерывания по совпадению канала A, обнуление регистров счета и контрольной суммы и установка адресного регистра на начало буфера приема, что соответствует полному обнулению результата работы приемника USART.

## Подпрограмма прерывания по завершению измерения АЦП

В данной подпрограмме происходит подсчет и сохранение суммы измерений АЦП. АЦП замеряет значение напряжение на выводе PA0 8 раз, затем считает их среднее арифметическое путем двоичного сдвига направо 3 раза.

# Листинг программы

; PB7, PB6, PB5, PB4 - выходы, управляющие драйвером

; PD0 - RXD0, PD1 - TXD0, PD2 - FLOW CONTROL

; PA0 - ADC0

.include "m164Adef.inc"

.def tempL = R16

.def tempH = R17

.def rxBytes = R18 ; Счетчик принятых байт

.def txBytes = R19 ; Счетчик переданных байт

.def rxChecksum = R20 ; Контрольная сумма принятых байт

.def flagReg = R21 ; Регистр флагов

.def ConvCount = R22 ; Счетчик измерений аналого-цифрового преобразователя

.def ADC1 = R23 ; Сумма значений АЦП

.def ADC2 = R24

.equ FLAG\_RECEIVE = 0 ; Флаг принятого запроса

.equ FLAG\_TRANSMIT = 1 ; Флаг завершения передачи

.equ FLAG\_ADC\_READY = 2 ; Флаг завершения замера АЦП

.equ VAL\_TX = 5 ; Количество байт для обмена

.equ VAL\_RX = 3

.equ VAL\_TIMEOUT = 35 ; Таймаут (35 переполнений - 1 секунда)

.equ VAL\_CONVERTER = 8 ; Количество измерений аналого-цифрового преобразователя

.equ VAL\_CONV\_SHIFT = 3 ; Сдвиг для среднего арифметического результатов

.equ PIN\_FLOW\_CTRL = 3 ; Пин flow control

.equ MCU\_ADDRESS = 1 ; Адрес микроконтроллера

.DSEG

varBuf\_Rx: .BYTE 8 ; Буфер приема

varBuf\_Tx: .BYTE 8 ; Буфер передачи

.CSEG

; Reset

.org $0000

rjmp Init

; Output Compare1A Interrupt Vector Address

.org $001A

rjmp RX\_Timeout

; UART Receive Complete Interrupt Vector Address

.org $0028

rjmp RX\_complete

; UART Data Register Empty Interrupt Vector Address

.org $002A

rjmp UDR\_empty

; UART Transmit Complete Interrupt Vector Address

.org $002C

rjmp TX\_complete

; Окончание считывания АЦП

.org $0030

rjmp ADC\_complete

Init:

; Инициализация стека (выбор вершины)

ldi tempL, LOW(RAMEND)

out SPL, tempL

ldi tempL, HIGH(RAMEND)

out SPH, tempL

; Инициализация портов ввода/вывода A

ldi tempL, 0b00000001

out DDRA, tempL

; Инициализация портов ввода/вывода B

; PB4-PB7 - выходы,

ldi tempL, 0b11110000

out DDRB, tempL

; Отключена подтяжка на всех выходах

ldi tempL, 0b00000000

out PORTB, tempL

; Инициализация портов ввода/вывода C

ldi tempL, 0b00111100

out DDRC, tempL

; Инициализация портов ввода/вывода D

; PD1, PD2 - выходы (TXD0, FLOW CONTROL), PD0 - вход (RXD0)

ldi tempL, 0b00000110

out DDRD, tempL

; Включение приема

sbi PORTD, PIN\_FLOW\_CTRL

; Инициализация UART

; Разрешение прерывания по завершению приема

; Установка режима межпроцессорного обмена

ldi tempL, (1<<MPCM0)

sts UCSR0A, tempL

; Установка режима приема данных

ldi tempL, (1<<RXCIE0)|(1<<UDRIE0)|(1<<RXEN0)|(1<<MPCM0)

sts UCSR0B, tempL

; Асинхронный режим, бит четности, 8 бит информации

ldi tempL, (1<<UPM00)|(1<<UCSZ00)|(1<<UCSZ10)

sts UCSR0C, tempL

; Частота тактирования - 8 МГц

; Частота обмена - 19200 бод, одинарная скорость

ldi tempL, 25

ldi tempH, 00

sts UBRR0L, tempL;

sts UBRR0H, tempH

; Инициализация таймера TCNT1

; Выходы OCnA и OCnB отключены

ldi tempL, 0

sts TCCR1A, tempL;

; Нет деления частоты, верхний предел счета - OCR1A

ldi tempL, (1<<WGM12)

sts TCCR1B, tempL

; Верхний предел счета = 0,006\*8000000=48000(0xBB80)

ldi tempH, 0xBB

ldi tempL, 0x80

sts OCR1AH, tempH

sts OCR1AL, tempL

; Инициализация аналого-цифрового конвертера

; Вход - ADC0, AREF включен

ldi tempL, 0

sts ADMUX, tempL

; Обнуление участка буфера Rx в SRAM

ldi YL, LOW(varBuf\_Rx)

ldi YH, HIGH(varBuf\_Rx)

ldi tempL, VAL\_RX

BufRxdNull:

st Y+, tempL

dec tempL

cpi tempL, 0x00

brne BufRxdNull

; Обнуление участка буфера Tx в SRAM

ldi YL, LOW(varBuf\_Tx)

ldi YH, HIGH(varBuf\_Tx)

ldi tempL, VAL\_TX

BufTxdNull:

st Y+, tempL

dec tempL

cpi tempL, 0x00

brne BufTxdNull

; Обнуление переменных

clr tempL

clr tempH

clr rxBytes

clr txBytes

clr rxChecksum

clr flagReg

; Разрешаем прерывания

sei

;==================================================

; Начало цикла программы

;==================================================

Start:

; Ожидание приема запроса

sbrc flagReg, FLAG\_RECEIVE

rjmp Received

rjmp Start

; Запрос принят

Received:

; Загрузка адреса устройства из буфера приема

ld tempL, Y+

; Загружаем информационный байт из буфера приема

ld tempL, Y+

; Очищаем все кроме четырех младших битов

andi tempL, 0x0F

; Выводим на порт

out PORTB, tempL

; Включение АЦП, разрешение прерывания по окончанию считывания, начало замера

ldi tempL, (1<<ADEN)|(1<<ADIE)|(1<<ADSC)

sts ADCSRA, tempL

; Ожидание окончания замеров

Wait\_ADC:

sbrc flagReg, FLAG\_ADC\_READY

rjmp Transmit

rjmp Wait\_ADC

; Формирование посылки для отправления

Transmit:

; Загрузка адреса начала буфера отправки в адресный регистр

ldi YL, LOW(varBuf\_Tx)

ldi YH, HIGH(varBuf\_Tx)

; Запись адреса устройства в первый байт

ldi tempL, MCU\_ADDRESS

st Y+, tempL

; Запись состояния портов ввода/вывода во второй байт

in tempL, PORTB

st Y+, tempL

; Запись результата измерения АЦП в третий и четвертый байты

st Y+, ADC2

st Y+, ADC1

; Вычисление контрольной суммы

ldi convCount, VAL\_TX

clr tempH

Checksum\_calc:

ld tempL, -Y

add tempH, tempL

dec convCount

cpi convCount, 1

breq Checksum\_end

rjmp Checksum\_calc

; Добавление контрольной суммы в последний байт

Checksum\_end:

com tempH

sts varBuf\_Tx + VAL\_TX - 1, tempH

; Flow control на передачу

cbi PORTD, PIN\_FLOW\_CTRL

; Заполнение регистра данных USART

ld tempL, Y+

sts UDR0, tempL

; Разрешение прерывания по окончанию приема и опустошению регистра данных

ldi tempL, (1<<TXEN1)|(1<<UDRIE1)

sts UCSR0B, tempL

End:

clr flagReg

rjmp Start

;==================================================

; Конец цикла программы

;==================================================

; Прерывание окончания приема по USART

RX\_complete:

; Сохранение регистров в стэк

push tempL

push tempH

in tempL, SREG

push tempL

; Загрузка принятой информации из регистра данных

RX\_UDR\_in:

lds tempL, UDR1

; Проверка статусного регистра USART на наличие ошибок

RX\_check:

lds tempH, UCSR1A

andi tempH, (1<<FE1)|(1<<DOR1)|(1<<UPE1)

breq RX\_no\_error

rjmp RX\_regs\_exit

; При отсутствии ошибок приема

RX\_no\_error:

st Y+, tempL

inc rxBytes

cpi rxBytes, 1

breq RX\_first\_byte

cpi tempH, VAL\_RX

breq RX\_checksum

add rxChecksum, tempL

rjmp RX\_regs\_exit

; При приеме первого байта

RX\_first\_byte:

; Проверка адреса

cpi tempL, MCU\_ADDRESS

brne RX\_regs\_exit

; Сброс бита межпроцессорного обмена

ldi tempL, 0

sts UCSR0A, tempL

; Добавление значения байта в контрольную сумму

add rxChecksum, tempL

lds tempL, TIMSK1

; Сброс таймера

clr tempH

sts TCNT1H,tempH

sts TCNT1L,tempH

; Разрешить прерывание по каналу A

bld tempL, OCIE1A

sts TIMSK1, tempL

rjmp RX\_regs\_exit

; Проверка контрольной суммы

RX\_checksum:

; Инверсия последнего принятого байта

com tempL

; Сравнение контрольных сумм

cp tempL, rxChecksum

brne RX\_end

; Установка флага успешного приема данных

ori flagReg, (1<<FLAG\_RECEIVE)

; Окончание приема данных

RX\_end:

; Запрет прерывания таймера по переполнению канала A

lds tempL, TIMSK1

andi tempL, ~(1<<OCIE1A)

sts TIMSK1, tempL

clr rxBytes

clr rxChecksum

; Установка указателя на начало буфера отправки

ldi YL, LOW(varBuf\_Tx)

ldi YH, HIGH(varBuf\_Tx)

; Загрузка регистров из стэка

RX\_regs\_exit:

pop tempL

out SREG, tempL

pop tempH

pop tempL

reti

; Прерывание по опустошению регистра данных

UDR\_empty:

; Сохранение регистров в стэк

push tempL

in tempL, SREG

push tempL

; Разрешение прерывания по завершению передачи

ldi tempL, (1<<TXEN1)|(1<<TXCIE1)

sts UCSR0B, tempL

; Загрузка адреса начала буфера передачи в адресный регистр

ldi YL, low(varBuf\_Tx)

ldi YH, high(varBuf\_Tx)

ld tempL, Y+

sts UDR0, tempL

; Загрузка регистров из стэка

pop tempL

out SREG, tempL

pop tempL

reti

; Прерывание по завершению приема

TX\_complete:

; Сохранение регистров в стэк

push tempL

in tempL, SREG

push tempL

push tempH

; Инкремент счетчика отправленных байтов

inc txBytes

; Проверка на последний байт

cpi txBytes, VAL\_TX

breq TX\_end

; Отправление данных

ld tempL, Y+

sts UDR1, tempL

rjmp TX\_regs\_exit

; Завершение отправки

TX\_end:

clr txBytes

; Установка режима межпроцессорного обмена

ldi tempL, (1<<MPCM0)

sts UCSR0A, tempL

; Разрешение прерывания по завершению приема

; Установка режима приема данных

ldi tempL, (1<<RXCIE1)|(1<<RXEN1)

sts UCSR0B, tempL

; Загрузка адреса начала буфера приема в адресный регистр

ldi YL, LOW(varBuf\_Rx)

ldi YH, HIGH(varBuf\_Rx)

ori flagReg, (1<<FLAG\_TRANSMIT)

; Flow control на прием

sbi PORTD, PIN\_FLOW\_CTRL

; Загрузка регистров из стэка

TX\_regs\_exit:

pop tempH

pop tempL

out SREG, tempL

pop tempL

reti

; Прерывание таймера по совпадению канала A

RX\_timeout:

; Сохранение регистров в стэк

push tempL

push tempH

in tempL, SREG

push tempL

; Запрет прерывания по совпадению канала A

lds tempL, TIMSK1

andi tempL, ~(1<<OCIE1A)

sts TIMSK1, tempL

; Обнуление регистров

clr rxBytes

clr rxChecksum

; Установка адресного регистра на адрес начала буфера приема

ldi YL, LOW(varBuf\_Rx)

ldi YH, HIGH(varBuf\_Rx)

; Загрузка регистров из стэка

pop tempL

out SREG, tempL

pop tempH

pop tempL

reti

; Прерывание по завершению измерения АЦП

ADC\_complete:

; Сохранение регистров в стэк

push tempL

push tempH

in tempL, SREG

push tempL

; Получение результатов измерения

lds tempL, ADCL

lds tempH, ADCH

; Сложение результатов с предыдущими результатами

add ADC1, tempL

adc ADC2, tempH

; Прибавление счетчика

inc ConvCount

; Достигнуто ли максимальное количество измерений

cpi ConvCount, VAL\_CONVERTER

breq ADC\_max\_measures

; Начало следующего замера

ldi tempL, (1<<ADEN)|(1<<ADIE)|(1<<ADSC)

sts ADCSRA, tempL

; Загрузка регистров из стэка

ADC\_regs\_exit:

pop tempH

pop tempL

out SREG, tempL

pop tempL

reti

; Достигнуто максимальное количество измерений

ADC\_max\_measures:

ldi ConvCount, VAL\_CONV\_SHIFT

clr tempL

; Вычисление среднего арифметического (деление на степень двойки сдвигом)

ADC\_middle:

lsr ADC2

brcc ADC\_mid\_skip

ldi tempL, 0b10000000

ADC\_mid\_skip:

lsr ADC1

or ADC1, tempL

dec ConvCount

cpi ConvCount, 0

breq ADC\_end

rjmp ADC\_middle

ADC\_end:

; Установка флага окончания вычисления значения АЦП

ori flagReg, (1<<FLAG\_ADC\_READY)

rjmp ADC\_regs\_exit