|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

Факультет: «Специальное машиностроение»

Кафедра: СМ11 «Подводные аппараты и роботы»

**Лабораторная работа № 3**

по курсу «Электронные устройства мехатронных и робототехнических систем»

Вариант 8

Выполнил: Ионин Даниил

Группа: СМ11-61Б

Проверил(а):

Москва, 2024 г.

ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 2](#_Toc166253696)

[Введение 3](#_Toc166253697)

[Тема работы: 3](#_Toc166253698)

[Цель работы: 3](#_Toc166253699)

[Условные обозначения 4](#_Toc166253700)

[Блок схема ATMEGA16 5](#_Toc166253701)

[Работа со стендом EasyAVR5 6](#_Toc166253702)

[Теоретическая часть 8](#_Toc166253703)

[Модуль USART 8](#_Toc166253704)

[Интерфейс RS232 (COM-порт) 9](#_Toc166253705)

[Протокол последовательной передачи данных 10](#_Toc166253706)

[Порядок обмена информацией. 11](#_Toc166253707)

[Практическая часть 13](#_Toc166253708)

[Постановка задачи 13](#_Toc166253709)

[Блок-схема программы 14](#_Toc166253710)

Введение

Тема работы:

Работа с универсальным синхронно-асинхронным передатчиком

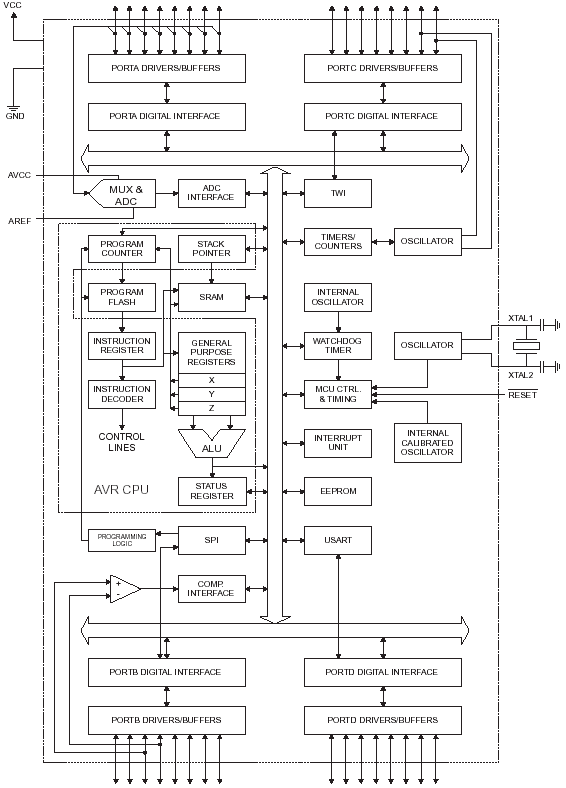
Цель работы:

Изучить работу универсального синхронно-асинхронного приемопередатчика. Изучить принципы сопряжения микроконтроллерных систем с ПК.

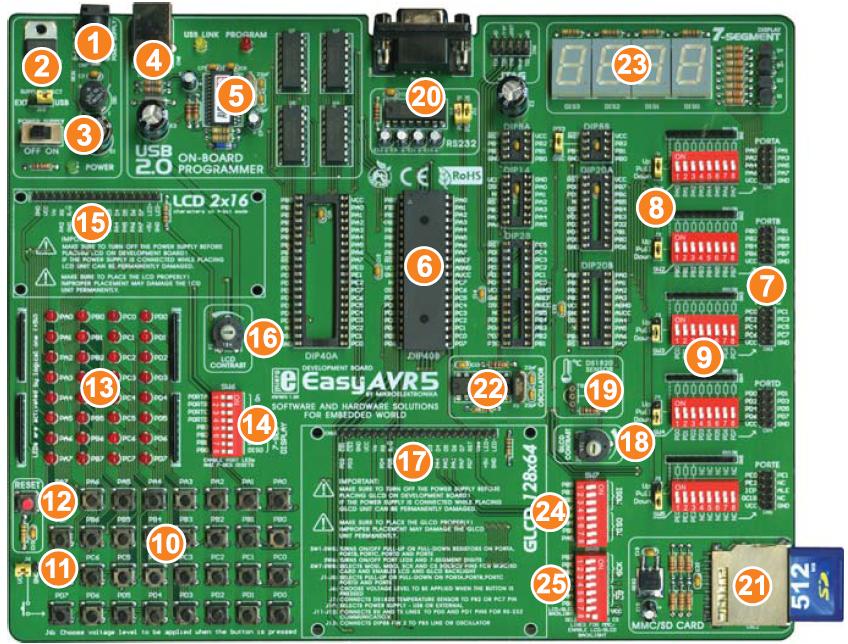
Условные обозначения

* Flash ROM - объем энергонезависимой памяти программ (в килобайтах);
* EEPROM - объем энергонезависимой памяти данных (в байтах);
* RAM - объем статической памяти данных (в байтах);
* External RAM - возможность подключения к микроконтроллеру дополнительной микросхемы внешней статической памяти данных (в килобайтах);
* ISP - возможность программирования микроконтроллера в системе (на целевой плате) при основном напряжении питания;
* SPM - функция самопрограммирования Flash ROM памяти микроконтроллера в системе без участия внешнего программатора;
* JTAG - встроенный JTAG - интерфейс;
* I/O (pins) - максимальное количество доступных линий ввода / вывода;
* Timer(s) 8/16 bit - количество и разрядность таймеров/счетчиков;
* USI - универсальный коммуникационный интерфейс;
* AC - аналоговый компаратор;
* ADC (channels) - количество каналов аналого-цифрового преобразования;
* Internal RC - наличие внутренней RC-цепочки для автономной работы микроконтроллера (без внешнего источника опорной частоты);
* WDT - сторожевой таймер;
* BDC - аппаратный программируемый блок защиты от сбоев при внезапном (в том числе и кратковременном) пропадании напряжения питания микроконтроллера;
* UART - асинхронный последовательный приемопередатчик;
* SPI - синхронный трехпроводной последовательный интерфейс;
* I2C - двухпроводной последовательный интерфейс;
* RTC - система реального времени;
* PWM (channels) - количество независимых каналов широтно - импульсной модуляции;
* Command Set - количество различных инструкций в системе команд микроконтроллера;
* Vcc - диапазон рабочих напряжений питания (в Вольтах);
* Clock - диапазон рабочих частот (в мегагерцах);

Блок схема ATMEGA16

 Рисунок 1 – Блок схема ATmega16

Работа со стендом EasyAVR5

 Рисунок 2 – Внешний вид стенда EasyAVR5

1 – разъем для подключения внешнего питания 8–16В;

2 – выбор между внешним питанием и питанием от USB;

3 – выключатель;

4 – разъем USB;

5 – встроенный внутрисхемный программатор;

6 – панельки для установки различных типов МК;

7 – разъемы для доступа к портам МК;

8 – джамперы для выбора типа подтягивающих резисторов (к 1 или к 0);

9 – группы переключателей для подключения подтягивающих резисторов к выводам МК;

10 – 32 кнопки, подключенные к выводам МК;

11 – джампер выбора высокого или низкого состояния при нажатии кнопки:

12 – кнопка сброса;

13 – светодиоды, каждый подключенный к выводу МК;

14 – группа переключателей, подключающая или отключающая светодиоды к МК;

15 – разъем для подключения символьной ЖК-матрицы;

16 – потенциометр для регулировки яркости символьной ЖК-матрицы;

17 – разъем для подключения графической ЖК-матрицы;

18 – потенциометр для регулировки яркости графической ЖК-матрицы;

19 – разъем для подключения датчика температуры DS1820;

20 – разъем порта RS232;

21 – слот для подключения MMC/SD карт памяти;

22 – встроенный генератор частоты;

23 – 4 семисегментных индикатора;

24 – группа переключателей для разрешения внутрисхемного программирования;

25 – группа переключателей для разрешения работы с MMC/SD картами памяти.

Теоретическая часть

Модуль USART

Через модуль USART (универсальный синхронно-асинхронный передатчик) микроконтроллер осуществляет прием и передачу информации, представленной последовательным кодом, поэтому его часто называют также последовательным портом. С помощью этого модуля микроконтроллер может обмениваться данными с различными внешними устройствами. Блок схема модуля USART представлена на рисунке 3.

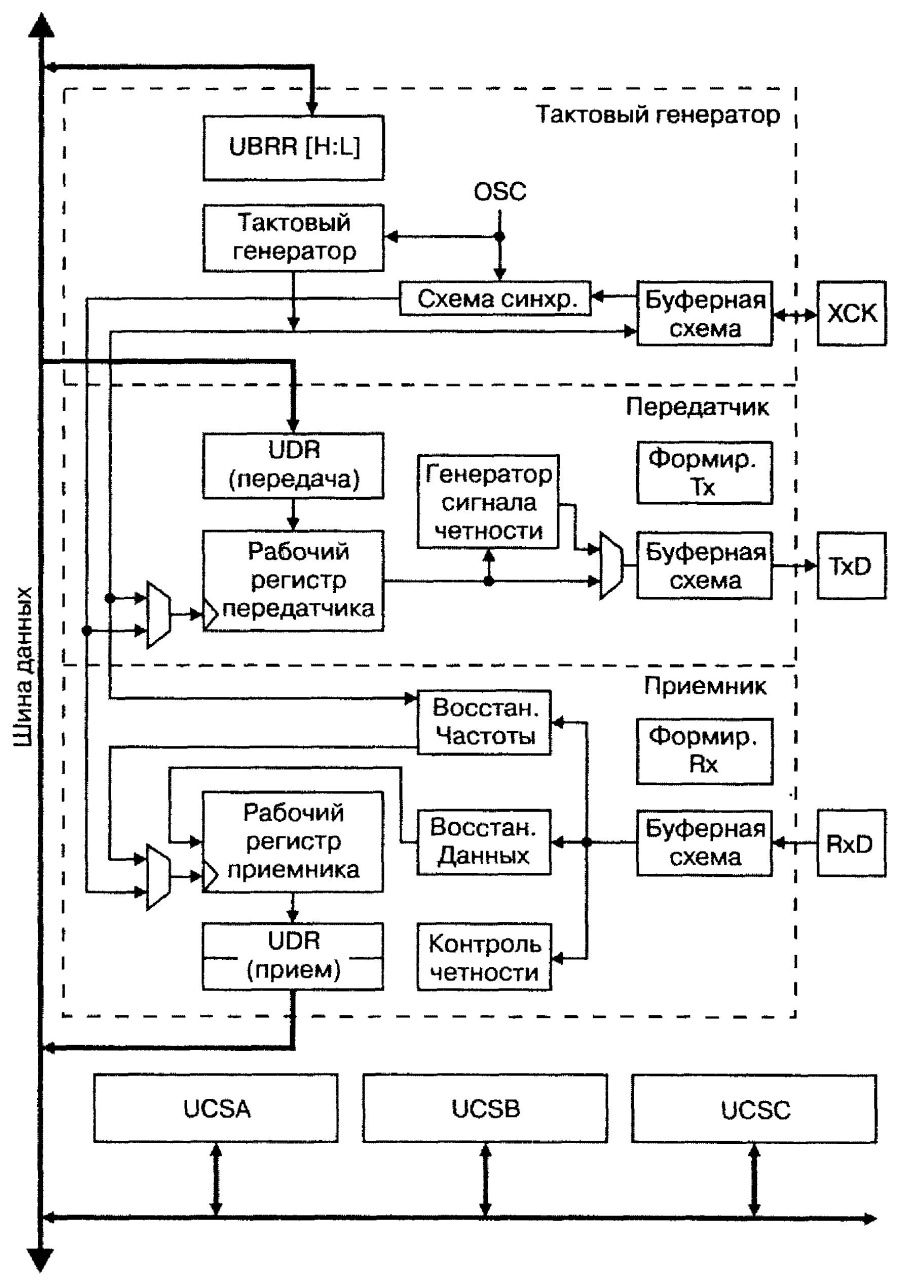


Рисунок 3 – USART

Скорость передачи данных может варьироваться в широких пределах, причем высокие скорости передачи могут быть достигнуты даже при относительно низкой тактовой частоте микроконтроллера.

Для взаимодействия с программой в модуле предусмотрены 3 раздельных прерывания, запрос на которые генерируется при наступлении следующих событий: «передача завершена», «регистр данных передатчика пуст» и «прием завершен».

Интерфейс RS232 (COM-порт)

Интерфейс RS-232C предназначен для подключения к компьютеру стандартных внешних устройств (принтера, сканера, модема, мыши и др.), а также для связи компьютеров между собой. Данные в RS-232C передаются в последовательном коде побайтно. Каждый байт обрамляется стартовым и стоповыми битами. Данные могут передаваться как в одну, так и в другую сторону (дуплексный режим).

Компьютер имеет 25-контактный (DB25P) или 9-контактный (DB9P) разъем для подключения RS-232C. Скорость передачи по RS-232C может выбираться из ряда: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 бит/с.

Все сигналы RS-232C передаются специально выбранными уровнями, обеспечивающими высокую помехоустойчивость связи (рисунок 7.). Отметим, что данные передаются в инверсном коде (логической единице соответствует низкий уровень, логическому нулю - высокий уровень).

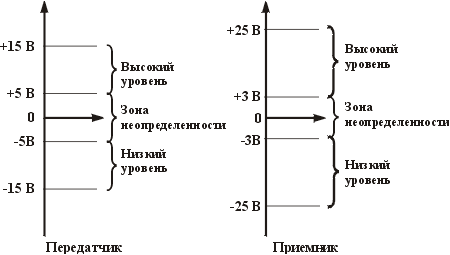


Рисунок 1 – Уровни сигналов RS-232C на передающем и принимающем концах линии связи

Для подключения произвольного УС к компьютеру через RS-232C обычно используют трех- или четырехпроводную линию связи, но можно задействовать и другие сигналы интерфейса. Для преобразования уровней сигналов с выхода модуля USART микроконтроллера к уровням интерфейса RS232 используются специальные микросхемы преобразования уровней (например, MAX232).

Протокол последовательной передачи данных

При передаче данных необходимо, чтобы имелся протокол обмена, согласно которому участвующие в обмене устройства будут интерпретировать данные. Кроме этого необходимо оговорить количество информационных бит в кадре данных, количество стоп-бит и т.д. В предлагаемой лабораторной работе принят следующий протокол обмена. Передача/прием данных осуществляется со следующими параметрами:

а) скорость обмена 19200/38400 бит/сек;

б) формат посылки 10/11 бит информации:

- старт-бит;

8 бит данных;

- бит контроля четности (может отсутствовать);

- один стоп бит.

Порядок обмена информацией.

С персонального компьютера, выполняющего роль ведущего в обмене (master), осуществляет передача кадра запроса выполнения простейшей арифметической операции. Кадр состоит из следующей последовательности:

1) Байт кода операции:

- 0x01 – сложение 2-х однобайтных двоичных чисел без знака (0 до 255);

- 0х02 – вычитание 2-х однобайтных двоичных чисел без знака (0 до 255);

- 0х03 – умножение 2-х однобайтных двоичных чисел без знака (0 до 255);

- 0х04 – деление 2-х однобайтных двоичных чисел без знака (0 до 255).

2) Байт первого операнда (1-ое слагаемое, уменьшаемое, множимое, делимое).

3) Байт второго операнда (2-ое слагаемое, вычитаемое, множитель, делитель).

4) Байт контрольной суммы с инверсией.

Микроконтроллер, ведомый (slave) принимает запрос на выполнение операции, осуществляет выполнение арифметической операции и возвращает кадр ответа выполненной операции. Кадр ответа состоит из следующей последовательности:

1) Старшего байта результата выполнения операции/(частное);

2) Младший байт результата выполнения операции/(остаток от деления);

3) Байт контрольной суммы с инверсией.

Передача запроса на вычисление и выдача ответа осуществляются асинхронно по нажатию соответствующих кнопок на экране монитора персонального компьютера и макетной платы.

Практическая часть

Постановка задачи

1. Выполните проверку работы программы в симуляторе AVR Studio. Обратите особое внимание проверке в симуляторе подпрограмм обработки прерываний Tx Complete, Data register Empty, RX Complete;

2. Выполните: программирование стенда EasyAVR5. Запустите программу AVRStend на ПК. Выполните передачу команды с ПК и прием ответа от стенда EasyAVR5. Проверьте правильность принятых и переданных данных на семисегментных индикаторах стенда EasyAVR5.

3. Доработайте программу сократив необоснованно большую паузу между нажатиями кнопки «Просмотр».

4. Проведите эксперимент по изменению и передаче команд арифметических операций и правильности их приема стендом EasyAVR5.

5. Измените скорость передачи в программе МК и программе AVRStend, проверьте и добейтесь правильной работы измененной программы.

6. Добавьте бит паритета в программах МК и AVRStend и проверьте, и добейтесь правильной работы измененной программы.

Полученный исправленный код находится в приложении 1.

Блок-схема программы