

## Лабораторная работа № 6. Последовательный обмен данными по каналу UART

### *Цель работы:*

- изучение структуры канала последовательного интерфейса UART (Universal Asynchronous Receiver-Transmitter),
- программирование передачи и приема данных по интерфейсу UART,
- моделирование и проверка работы последовательного канала в режиме обмена между двумя микроконтроллерами.

### **Введение**

Микроконтроллеры AVR имеют в своём составе модуль полнодуплексного универсального асинхронного приёмопередатчика UART (в семействе Mega универсальный синхронный/асинхронный приёмопередатчик USART). Через него осуществляется приём и передача информации, представленной последовательным кодом, поэтому модуль UART часто называют также последовательным портом. С помощью этого модуля микроконтроллер может обмениваться данными с различными внешними устройствами.

Поток данных, передаваемых по каналу UART, представляет собой совокупность посылок или кадров. Каждый кадр содержит стартовый бит, восемь или девять битов данных и стоповый бит. Стартовый бит имеет уровень логического 0, стоповый – уровень логической 1.

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

Известно, что при передаче данных могут происходить различные сбои. Модуль UART, реализованный в микроконтроллерах, может обнаруживать переполнение и ошибку формата.

Для взаимодействия с программой в модуле предусмотрены прерывания при наступлении следующих событий: «приём завершён» с адресом вектора \$009 в таблице векторов прерываний, «регистр данных передатчика пуст» с адресом вектора \$00A, «передача завершена» с адресом вектора \$00B.

Выводы микроконтроллера, используемые модулем UART, являются линиями порта PD. В качестве входа приёмника (RXD) используется вывод PD0, а в качестве выхода передатчика (TXD) – вывод PD1.

Принимаемые и передаваемые данные (8 разрядов) хранятся в регистре UDR. Физически регистр UDR состоит из двух отдельных регистров, один из которых используется для передачи

данных, другой – для приёма. При чтении регистра UDR выполняется обращение к регистру приёмника, при записи – к регистру передатчика.

### Управление работой UART

Управление работой приёмопередатчика осуществляется с помощью регистра управления UCR. Текущее состояние приёмопередатчика определяется с помощью регистра состояния USR. Формат этих регистров приведен в таблицах 1 и 2.

Таблица 1. Формат регистра управления UCR

№ разряда	7	6	5	4	3	2	1	0
Имя	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8

Назначение управляющих битов:

RXCIE - разрешение прерывания по завершении приёма, если RXCIE=1;

TXCIE - разрешение прерывания по завершении передачи, если TXCIE=1;

UDRIE - разрешение прерывания при опустошении регистра данных UART, если UDRIE=1;

RXEN - разрешение приёма при RXEN=1;

TXEN - разрешение передачи при TXEN=1;

CHR9 - формат посылок. Если CHR9=1, производится передача и приём 9-разрядных данных. При передаче значение старшего (8-го) разряда берётся из разряда TXB8 регистра, а при приёме записывается в разряд RXB8;

RXB8 - 8-й разряд принимаемых данных. Если флаг CHR9=1, разряд RXB8 содержит значение старшего разряда принятого слова;

TXB8 - 8-й разряд передаваемых данных. Если флаг CHR9=1, значение TXB8 передаётся как старший разряд слова.

Таблица 2. Формат регистра состояния USR

№ разряда	7	6	5	4	3	2	1	0
Имя	RXC	TXC	UDRE	FE	OR	-	-	-

Значения битов (флагов) регистра состояния:

RXC - флаг завершения приёма. Данный флаг устанавливается в «1» при пересылке принятого слова из сдвигового регистра приёмника в регистр данных UDR. Сбрасывается флаг аппаратно при чтении регистра UDR;

TXC – флаг завершения передачи. Данный флаг устанавливается в «1» после передачи всех разрядов слова, включая стоп-бит, из сдвигового регистра передатчика, при условии, что в регистр данных UDR не было загружено новое значение. Этот флаг наиболее полезен при полудуплексной связи, при которой передающее устройство должно освободить линию и перейти в режим приёма

сразу же после окончания передачи. Флаг сбрасывается аппаратно при выполнении подпрограммы обработки прерывания или программно при записи «1» (!) в этот разряд;

UDRE – регистр данных пуст. Данный флаг устанавливается в «1» после пересылки байта из регистра данных UDR в сдвиговый регистр передатчика. Установка этого флага означает, что передатчик готов к получению нового значения для передачи. Сбрасывается флаг аппаратно при записи в регистр UDR;

FE – флаг ошибки формата. Данный флаг устанавливается в «1», если стоп-бит принятого слова равен «0». Флаг сбрасывается при приёме стоп-бита, равного «1»;

OR - флаг переполнения. Данный флаг устанавливается в «1», если в сдвиговом регистре приёмника находится новое принятое слово, а старое содержимое регистра UDR не прочитано. Флаг остаётся установленным до тех пор, пока не будет прочитано содержимое регистра UDR. Флаг сбрасывается при пересылке принятых данных из сдвигового регистра приёмника в регистр UDR

Напомним, что в микроконтроллерах ATmega8515 функции регистров UCR, USR и UBRR выполняют соответственно регистры UCSRB, UCSRA, UBRRH:UBRRL. Заметим, что в регистре UCSRA имеется дополнительный разряд MPCM, присутствие которого позволяет организовать простейшую локальную микроконтроллерную систему. При значении MPCM=1 приемник принимает кадры, у которых 9-й бит равен 1 и игнорирует кадры с 9-м битом, равным 0. При значении MPCM=0 приемник принимает кадры с любым значением 9-го бита. В микроконтроллерной сети один контроллер имеет статус ведущего, остальные – статус ведомых. У всех ведомых контроллеров перед началом работы бит MPCM=1. Ведущий контроллер посылает байт, содержащий адрес ведомого контроллера, с 9-м битом, равным 1. Все контроллеры принимают его, проверяют поступивший адрес, сравнивая его с собственным адресом, присвоенным при инициализации сети. Тот контроллер, который опознал поступивший адрес как собственный, переключает бит MPCM в состояние “0”. Данные, пересылаемые ведущим контроллерам, содержат 9-й бит, равный 0. Эти данные могут быть приняты только тем контроллером, бит которого MPCM=0. Если от ведущего контроллера поступит новый адрес в кадре с 9-м битом, равным 1, принимавший контроллер изменит значение MPCM на 1 и вернется в исходное состояние.

### **Передача данных**

Структурная схема передатчика модуля UART приведена на рис. 1.

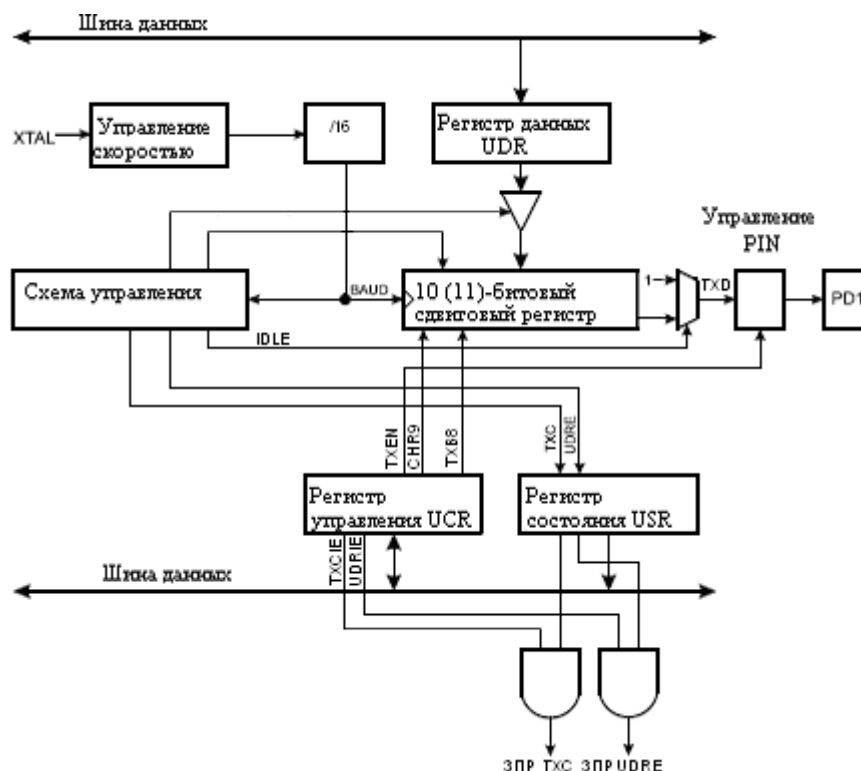


Рис. 1. Структурная схема передатчика UART

Работа передатчика разрешается установкой в «1» разряда TXEN регистра UCR. Если этот разряд сброшен (передатчик выключен), вывод PD1 (TXD) может использоваться как линия ввода/вывода порта PD. При установке разряда TXEN этот вывод подключается к передатчику UART и начинает функционировать как выход независимо от состояния 1-го разряда регистра DDRD порта PD.

Передача инициируется записью передаваемых данных в регистр данных UDR. После этого данные пересылаются из регистра UDR в сдвиговый регистр передатчика. При этом возможны два варианта:

- 1) новое значение записывается в регистр UDR после того, как был передан стоп-бит предыдущего слова. В этом случае данные пересылаются в сдвиговый регистр сразу же после записи в регистр UDR;
- 2) новое значение записывается в регистр UDR во время передачи. В этом случае данные пересылаются в сдвиговый регистр после передачи стоп-бита текущего слова.

После пересылки содержимого регистра UDR в сдвиговый регистр флаг UDRE регистра USR устанавливается в «1», что означает готовность передатчика к получению нового значения. В этом состоянии флаг остаётся до новой записи в регистр UDR. Одновременно с пересылкой формируется служебная информация: 0-й разряд сдвигового регистра сбрасывается в «0» (старт-бит), а 9-й (или 10-й) разряд устанавливается в «1» (стоп-бит). Если включен режим передачи 9-разрядных данных

(разряд CHR9 регистра UCR установлен в «1»), то значение разряда TXB8 регистра UCR копируется в 9-ый разряд сдвигового регистра.

После загрузки сдвигового регистра его содержимое начинает сдвигаться вправо и поступает на вывод TXD в следующем порядке: стартовый бит, данные (начиная с младшего разряда), стоповый бит. Сдвиг осуществляется по тактовому сигналу, вырабатываемому контроллером скорости передачи. Если во время передачи в регистр UDR было записано новое значение, то после передачи стоп-бита оно пересылается в сдвиговый регистр. Если же к моменту окончания передачи стоп-бита такой записи выполнено не было, в регистре USR устанавливается флаг завершения передачи TXC.

### Приём данных

Структурная схема приёмника модуля UART приведена на рис. 2. После обнаружения стартового бита начинается обработка поступающих разрядов слова данных. Решение о значении принятого разряда принимается по результатам 3-х выборок входного сигнала в середине битового периода. Состоянием разряда считается логическое значение, которое было получено, по меньшей мере в двух из трёх выборок. По мере распознавания разрядов принимаемой последовательности они поступают, сдвигаясь вправо, в сдвиговый регистр приёмника.

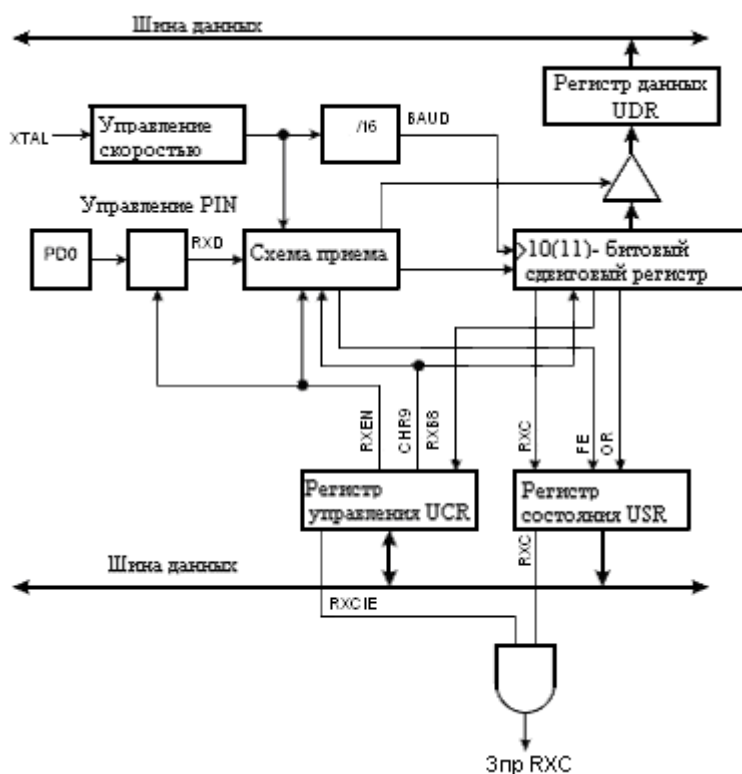


Рис. 2. Структурная схема приёмника UART

Распознавание стоп-бита производится также по трём выборкам входного сигнала. Стоп-бит считается принятым, если значение хотя бы двух из трёх выборок входного сигнала равно «1». В

противном случае фиксируется ошибка кадра, и флаг FE регистра USR устанавливается в «1». Перед чтением регистра данных UDR следует всегда проверять состояние этого флага.

Независимо от того был или не был обнаружен стоп-бит в конце принимаемой посылки, принятое слово пересылается в регистр данных UDR и устанавливается флаг RXC регистра USR. В случае обмена 9-разрядными данными при пересылке содержимого сдвигового регистра приёмника в регистр данных 9-й разряд принятого слова загружается в разряд RXB8 регистра UCR.

Если новое слово будет принято до того, как из регистра UDR будут считаны предыдущие данные, возникает переполнение. Об этом сигнализирует флаг OR регистра USR, который в этом случае устанавливается в «1». Установка этого флага означает, что принятые данные не могут быть переданы из сдвигового регистра в регистр данных и оказываются потерянными. Сбрасывается указанный флаг только после обращения к регистру данных. Поэтому при высоких скоростях передачи либо большой загрузке процессора программа должна проверять состояние флага OR в регистре USR для обнаружения возможного переполнения.

### **Скорость приёма/передачи**

Управление скоростью приёма и передачи данных осуществляется контроллером скорости передачи, который является обыкновенным делителем частоты. Скорость передачи зависит от содержимого регистра контроллера. В AT90S8515 это регистр ввода/вывода UBRR, в ATmega8515 – регистры UBRRH:UBRRL Скорость передачи определяется следующим выражением:

$$BAUD = \frac{f_{CLK}}{16 \cdot (UBRR + 1)},$$

где BAUD – скорость передачи (в бодах),

$f_{CLK}$  - тактовая частота микроконтроллера (Гц),

UBRR - содержимое регистра контроллера скорости передачи.

Как известно, существует ряд значений скорости передачи данных, являющихся, по сути дела, стандартными. Значения UBRR, позволяющие получить эти скорости передачи при использовании различных резонаторов, а также величины ошибок получаемых значений относительно их теоретического значения приведены в табл. 3. Поскольку максимальная тактовая частота, устанавливаемая с помощью STK500, равна 3,69 МГц, значения частот выше этой величины в табл. 3 не приведены. Так как при увеличении ошибки помехозащищённость линии передачи снижается, скорости передачи, имеющие ошибку установки более 1%, использовать не рекомендуется.

Таблица 3. Значения UBRR для различных значений  $f_{CLK}$ 

Скорость (бод)	$f_{CLK}=1$ МГц	Ошибка, %	$f_{CLK}=2.4576$ МГц	Ошибка, %	$f_{CLK}=3.6864$ МГц	Ошибка, %
2400	25	0.2	63	0	95	0
4800	12	0.2	31	0	47	0
9600	6	7.5	15	0	23	0
14400	3	7.8	10	3.1	15	0
19200	2	7.8	7	0.0	11	0
28800	1	7.8	4	6.3	7	0
38400	1	22.9	3	0	5	0
57600	0	7.8	2	12.5	3	0
76800	0	22.9	1	0	2	0
115200	0	84.3	0	25.0	1	0

### Практическая часть

**Задание 1.** Подготовить программы для исследования передачи и приёма по последовательному каналу UART. Напомним, что в качестве выхода передатчика используется вывод PD1 (TXD), а в качестве входа приёмника – вывод PD0 (RXD).

Визуально, используя светодиодную индикацию, контролировать передаваемые данные невозможно из-за большой скорости передачи. Поэтому возможны следующие варианты контроля работы канала UART:

- 1) с помощью программ симуляторов;
- 2) используя программу Hiper Terminal для обмена сообщениями через COM – порт персонального компьютера с микроконтроллером на плате STK500 по интерфейсу RS-232;
- 3) соединив выводы UART двух микроконтроллеров, размещенных на платах STK500, напрямую, либо через разъем интерфейса RS-232, и подключив к выходам принимающего микроконтроллера светодиодную индикацию STK500. Схема включения микроконтроллеров МК1 и МК2 для этого случая приведена на рис. 3.

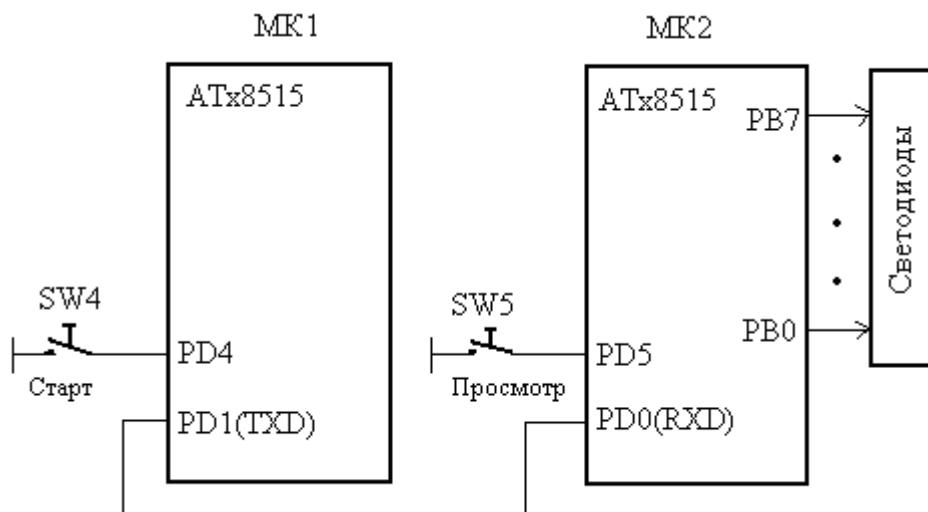


Рис. 3. Схема устройства для передачи сообщения по интерфейсу UART

Используем микроконтроллер MK1 для передачи данных, микроконтроллер MK2 - для приема. Данные для передачи, три байта сообщения, разместим в памяти программ MK1. Передача начинается с нажатия кнопки SW4 (Старт). Первый микроконтроллер выполняет в цикле последовательный вывод данных. После приема каждого байта данных второй микроконтроллер сохраняет их в своей памяти SRAM, начиная с адреса \$180. Закончив прием, второй микроконтроллер последовательно выводит при нажатии кнопки SW5 (Просмотр) полученные данные на светодиоды.

Схемы алгоритмов передачи и приема приведены на рис. 4.



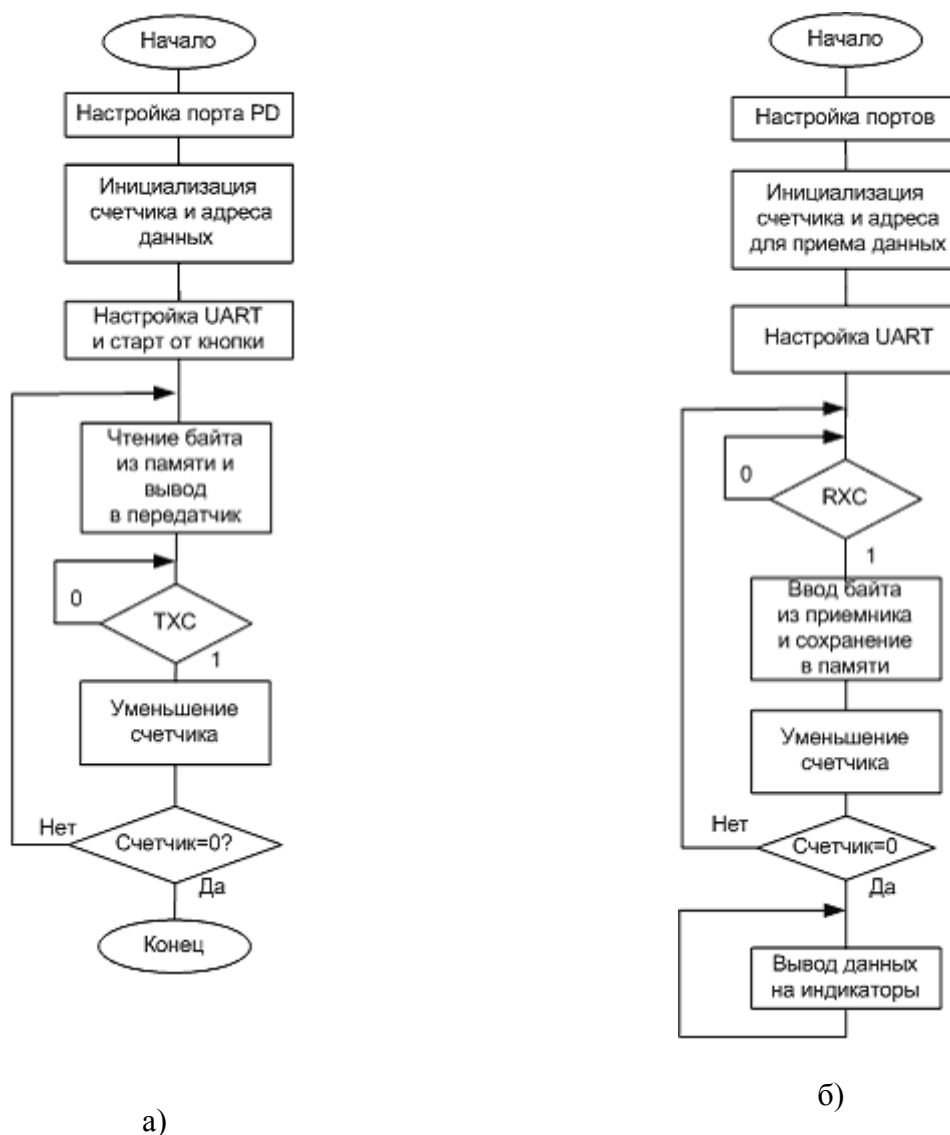


Рис. 4. Схемы алгоритмов передачи из микроконтроллера 1 (а) и приема в микроконтроллер 2 (б)

После инициализации оба микроконтроллера переходят в режим ожидания нажатия кнопки SW4. Микроконтроллер МК2, программируемый для приема данных, ожидает поступления данных. Его дальнейшая работа зависит от значения флага RXC: ожидание - при RXC=0, и ввод байта данных - при RXC=1. Состояние этого флага зависит от работы передающего устройства. Переход к последующим итерациям в циклах передачи (приема) происходит после установки флага завершения передачи TXC (соответственно приема RXC) очередного байта данных. После завершения цикла приёма включаются все светодиоды линейки индикации. Далее при последовательном нажатии на кнопку SW5 (Просмотр) осуществляется вывод принятых байтов данных на светодиоды. Ниже приведены тексты программ для передающего и принимающего микроконтроллеров.

**Программа 6.1**

```

;*****
;Программа 6.1 для МК АТх8515: демонстрация работы UART
;При нажатии на SW4 (START) происходит последовательная передача
;по каналу UART трёх байтов сообщения, считываемых из ячеек flash-памяти.
;Частота тактового генератора = 3,69 МГц,
;при UBRRL=11 скорость передачи 19219 бод
;Соединения: PD4-SW4, PD1-TXD (PD0-RXD)
;*****

.include "8515def.inc"      ;файл определений AT90S8515
.include "m8515def.inc"    ;файл определений ATmega8515
.def temp = r16             ;временный регистр
.def count = r17            ;счётчик
.equ START = 4              ;4-й вывод порта PD

.org $000

        rjmp init

;***Инициализация МК
INIT:    ldi ZL,low(text*2)  ;загрузка адреса текста
        ldi ZH,high(text*2) ;сообщения в регистр Z
        ldi count,3         ;установка счётчика байтов
        clr temp            ;настройка
        out DDRD,temp       ; вывода
        ldi temp,0x10       ; порта PD4
        out PORTD,temp      ; на ввод

;***Настройка UART на передачу данных
;/// для AT90S8515 регистр UCR вместо UCSRB и UBRR
        ldi temp,0x08       ;разрешение
        out UCSRB,temp      ; передачи по каналу UART
        ldi temp,11         ;скорость передачи для UBRRL (UBRR)
        out UBRRL,temp      ; 19219 бод
WAIT_START:sbic PIND,START  ;ожидание нажатия
        rjmp WAIT_START    ; кнопки START
OUTPUT:  lpm                 ;считывание байта из flash-памяти в r0
        out UDR,r0         ;вывод байта в передатчик

;/// для AT90S8515 регистр USR вместо UCSRA
        sbi UCSRA,TXC       ; сброс флага TXC
WAIT:    sbic UCSRA,TXC     ;ожидание

```

```

        rjmp next          ; завершения
        rjmp WAIT          ; передачи
next:    adiw z1,1          ;увеличение указателя адреса на 1
        dec count          ;уменьшение счётчика на 1
        brne OUTPUT        ;продолжение вывода
fin:     rjmp fin           ;передача завершена
text:    .db 'A','V','R'    ;текст сообщения (коды $41,$56,$52)

```

## Программа 6.2

```

;*****
;Программа 6.2 для МК ATx8515: демонстрация работы канала UART
;в режиме приема трёх байтов. Частота тактового генератора = 3,69 МГц,
;при UBRRL=11 скорость обмена 19219 бод
;Соединения: шлейфом порт PB-LED, PD5-SW5, PD0-RXD (PD1-TXD)
;*****
;.include "8515def.inc"          ;файл определений AT90S8515
;.include "m8515def.inc"        ;файл определений ATmega8515
.def temp = r16                  ;временный регистр
.def count = r17                 ;счётчик
.equ SHOW = 5                    ;5-й вывод порта PD

.org $000

        rjmp init
;***Инициализация МК
INIT:    ldi temp,low(RAMEND)     ;установка
        out SPL,temp            ; указателя стека
        ldi temp,high(RAMEND)    ; на последнюю
        out SPH,temp            ; ячейку ОЗУ
        ldi YL,0x80              ;в регистре Y – адрес, по которому
        ldi YH,0x01              ; происходит запись принятых данных
        ldi count,3              ;установка счётчика байтов
        ser temp                 ;настройка
        out DDRB,temp            ; порта PB на вывод
        out PORTB,temp           ; и выключение светодиодов
        clr temp
        out DDRD,temp            ;настройка
        ldi temp,0x20            ; вывода PD5

```

```

        out PORTD,temp                ; на ввод
;***Настройка UART на приём данных
        ;/// для AT90S8515 регистр UCR вместо UCSRB и UBRR
        ldi temp,0x10                ;разрешение приёма
        out UCSRB,temp               ; по каналу UART
        ldi temp,11                  ;скорость приёма/передачи
        out UBRRL,temp               ; 19219 бод
        ;/// для AT90S8515 регистр USR вместо UCSRA
WAIT_RXC: sbic UCSRA,RXC              ;ожидание
        rjmp INPUT                   ; завершения
        rjmp WAIT_RXC                ; приёма
INPUT:   in temp,UDR                  ;ввод байта из приёмника
        st Y+,temp                   ;и сохранение в памяти
        dec count                    ;уменьшение счётчика на 1
        brne WAIT_RXC                ;продолжение приема
        clr temp                     ;сигнализация -
        out PORTB,temp               ; приём завершен
LOOP:    ldi YL,0x80                  ;установка начального адреса
        ldi count,3                  ;установка счётчика байтов
WAIT_SHOW: sbic PIND,SHOW              ;ожидание нажатия
        rjmp WAIT_SHOW                ; кнопки SW5
        ld temp, Y+                  ;считывание байта из памяти
        com temp                      ;инвертирование
        out PORTB,temp               ;вывод на светодиоды
        rcall DELAY                  ;задержка
        dec count                    ;если показаны не все данные,
        brne WAIT_SHOW                ; то продолжение при нажатии SW5
        ser temp                      ;вывод окончен
        out PORTB,temp               ;светодиоды погашены
        rjmp LOOP                    ;повторение вывода

;*** Задержка ***
DELAY:   ldi r19,20
        ldi r20,255
        ldi r21,255
dd:      dec r21
        brne dd
        dec r20

```

```
brne dd  
dec r19  
brne dd  
ret
```

***Проверка работы канала передачи в режиме отладки.***

Создать в AVR Studio 4 проект для передачи данных с помощью программы 6.1. Проверить работу программы в режиме отладки, наблюдая состояния регистров и битов состояния канала UART и контролируя состояние линии PD1/TxD. С помощью системных часов *Stop Watch* измерить длительность одного бита данных на линии PD1 и оценить скорость передачи.

Отлаженную программу запрограммировать в память микроконтроллера на плате STK500. Закрывать программу AVR Studio 4.

***Проверка работы канала UART с помощью программы Hyper Terminal.***

Проверить работу канала передачи UART, соединив STK500 с компьютером. Для этого используем интерфейс RS-232 для передачи/приема данных из микроконтроллера по каналу UART. Подключение микроконтроллера к драйверу интерфейса RS-232 выполняется 2-проводным шнуром путем соединения выводов порта PD0, PD1 с контактами RXD, TXD 2-контактного разъема “RS232 SPARE”. Соединение платы с компьютером можно осуществить при выключенном напряжении питания на плате кабелем связи через 9-контактный разъем интерфейса RS-232, обозначенный на плате “RS232 SPARE”. Соединить кнопку SW4 (Старт) с входом PD4. После соединения подать напряжение питания на плату.

Запустить программу Hyper Terminal из Windows (либо альтернативную), выбрав одноименную строку в меню Программы/ Стандартные/ Связь, и настроить канал связи, установив COM - порт, скорость обмена 19200 бод, 8-битовый формат. На плате замкнуть кнопку SW4 для запуска передачи сообщения. Просмотреть в окне программы Hyper Terminal компьютера принятое сообщение.

Повторить передачу, инициируя повторную работу программы кнопкой RESET и выполняя передачу от кнопки Старт.

***Проверка работы приемника UART в режиме отладки.***

Создать в AVR Studio 4 проект для приема данных с помощью программы 6.2. Проверить работу программы в режиме симуляции, “вручную” устанавливая флаг приема RXC и данные в регистре UDR и контролируя прием данных во внутреннюю память SRAM и затем вывод в выходной порт PB путем эмуляции замыкания кнопки Просмотр (устанавливая 0 в разряде PIND5) .

Отлаженную программу запрограммировать в память микроконтроллера на плате STK500. Закрывать программу AVR Studio 4.

### ***Проверка работы канала UART с помощью программы Hyper Terminal.***

Подготовить текстовый файл короткого сообщения (из 3-х символов) с помощью стандартной программы Блокнот. Выполнить соединения, как описано выше. Включить напряжение питания на плате. Снова запустить программу Hyper Terminal и настроить канал связи (скорость передачи 19200 бит/с, биты данных - 8, четность - нет, стоповые биты - 1, управление потоком: Xon/Xoff). Выполнить команды меню Передача/ Отправить текстовый файл. После включения светодиодов на плате (признак приема 3-х символов) вывести на индикаторы ASCII - коды принятых символов, трижды нажимая кнопку SW5.

Перезапустив программу приема с помощью кнопки RESET, выполнить передачу набора цифровых данных непосредственно с клавиатуры.

### ***Проверка работы канала UART при передаче данных между микроконтроллерами.***

А) Запустить программу Proteus ISIS. Создать проект для схемы на рис. 3. Добавить файлы программ передатчика и приемника. Выполнить компиляцию.

В окне редактирования свойств микроконтроллера (Edit Properties), которое открывается после выделения микроконтроллера правой кнопкой мыши и выбора соответствующей строки меню, установить необходимую частоту синхронизации микроконтроллера. Для этого необходимо выбрать опцию CKSEL Fuses - (0111) Ext.RC (3.0MHz -8.0 MHz) и в нижерасположенной строке указать конкретное значение частоты Clock Frequency. Повторить операцию настройки для второго микроконтроллера.

Проверить работу каналов UART в режиме отладки:

- подключив виртуальный терминал к линии передачи (Virtual Instruments Mode/ VIRTUAL TERMINAL) настроить его для приема передаваемого сообщения, открыв для этого окно свойств, и выполнить передачу сообщения, наблюдая в окне терминала полученные данные (открыть окно терминала можно, выполнив команды меню Debug/Virtual terminal);

- разведив микроконтроллеры, подключить виртуальный терминал к входу приемника и записав в окне терминала пересылаемые данные проверить их получение, проверив внутреннюю память данных микроконтроллера и контролируя выходы порта PB при нажатии кнопки просмотра SW5;

- восстановить связь между микроконтроллерами и выполнить передачу сообщения из МК1 в МК2;

- открыть окно для отображения временной диаграммы Digital Analysis из Graph Mode, подключить пробник напряжения к линии передачи. Щелкнув правой кнопкой мыши в окне, отредактировать трассу и временной интервал для отображения временной диаграммы передачи трех символов. Для исключения времени ожидания нажатия кнопки Старт при записи временной

диаграммы ее следует перед выполнением симуляции либо замкнуть, либо закомментировать в программе ее проверку, заново выполнив компиляцию. Выполнить моделирование и расшифровать временную диаграмму.

Б) Для проверки работы канала UART при передаче сообщения из микроконтроллера МК1 в микроконтроллер МК2 необходимо после программирования микроконтроллеров (!) соединить напрямую вывод PD1(TXD) первого STK500 с выводом PD0 второго, а также соединить выводы GND обоих плат. При необходимости платы STK500 после программирования можно отсоединить от кабелей RS-232. При нажатии кнопок RESET оба микроконтроллера переводятся в режим ожидания. Передача сообщения начинается после нажатия кнопки SW4 (Старт) на плате первого STK500. После приема сообщения загораются все светодиоды второго STK500, показывая, что прием данных завершен. Полученные данные можно вывести на индикаторы, нажимая последовательно кнопку Просмотр (SW5).

**Задание 2.** Написать программу для передачи сообщения “hello”, хранимого в памяти программ микроконтроллера STK500-1, в память данных микроконтроллера STK500-2.

Задание (факультативное). Одну из процедур (передачи или приема) запрограммировать, используя запросы прерываний (от передатчика или приемника).

### Оформление отчета

Отчет должен содержать:

- а) упрощенные структурные схемы передатчика и приемника с указанием программно-доступных регистров и их назначением,
- б) программы для передачи и приема данных по интерфейсу UART,
- в) схему проекта для обмена данными между двумя микроконтроллерами (“скриншот” из окна редактора Proteus) с результатами, подтверждающими работу передатчика и приемника (содержимое памяти данных приемника, окно терминала),
- г) осциллограмму и расшифровку сигналов на линии передачи TxD-RxD,
- д) выводы по работе программ при передаче данных с платы на плату.

Требования при защите: уметь отвечать на вопросы по работе схемы, кодам программ, методике выполнения эксперимента.

### Контрольные вопросы

1. Как осуществляется асинхронный обмен данными по последовательному интерфейсу? В каком порядке следуют служебные биты и биты данных при выводе байта?

2. Как организован вывод в передатчик сообщения из программной Flash-памяти микроконтроллера? Какую роль выполняет регистр Z и почему при его начальной загрузке удваивается значение переменной *text*?
3. Почему светодиоды на плате зажигаются после приема заданного количества символов?
4. Почему время индикации последнего принятого символа оказывается короче?
5. В какое состояние переходит программа после передачи и после просмотра принятого сообщения? Каким образом следует перезапустить передачу-прием?
6. Рассчитайте период представления битов на линии TXD при передаче данных со скоростью 19200 бит/с.
7. Как установить в симуляторах ISIS Proteus (и в AVR Studio 4) частоту работы микроконтроллера?
8. Как остановить прием сообщения, длина которого заранее неизвестна?
9. Написать программу, осуществив передачу данных из памяти SRAM по каналу UART. Загрузку данных (констант) в ячейки SRAM осуществить непосредственно командами программы.

#### Рекомендуемая литература

1. В.Я. Хартов Микроконтроллеры AVR. Практикум для начинающих. Издательство МГТУ им. Н.Э. Баумана. 2012 г.