



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 ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

## О Т Ч Е Т

по лабораторной работе № 6

**Название:** Последовательный обмен данными по каналу UART

**Дисциплина:** Микропроцессорные системы.

Студент

ИУ6-62Б

(Группа)

(Подпись, дата)

С.В. Астахов, Д.И. Вариханов

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

(И.О. Фамилия)

Москва, 2022

## Вариант 1.

### Цели 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.

### Ход работы.

#### Задание 1

Подготовить программы для исследования передачи и приёма по последовательному каналу UART.

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.

Проверить работу программы.

Структурная схема передатчика и приемника модуля UART приведены на рисунках 1 и 2 соответственно.

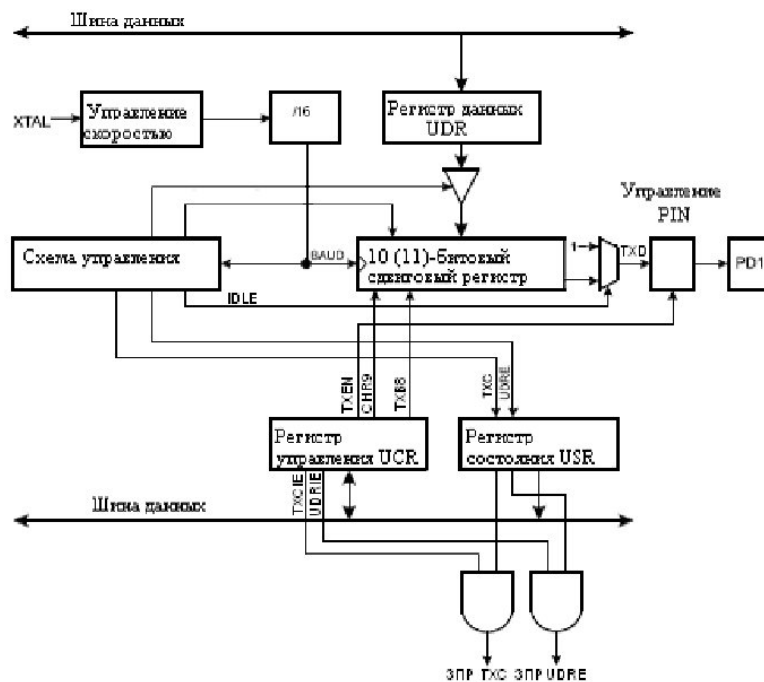


Рисунок 1 - структурная схема передатчика модуля UART

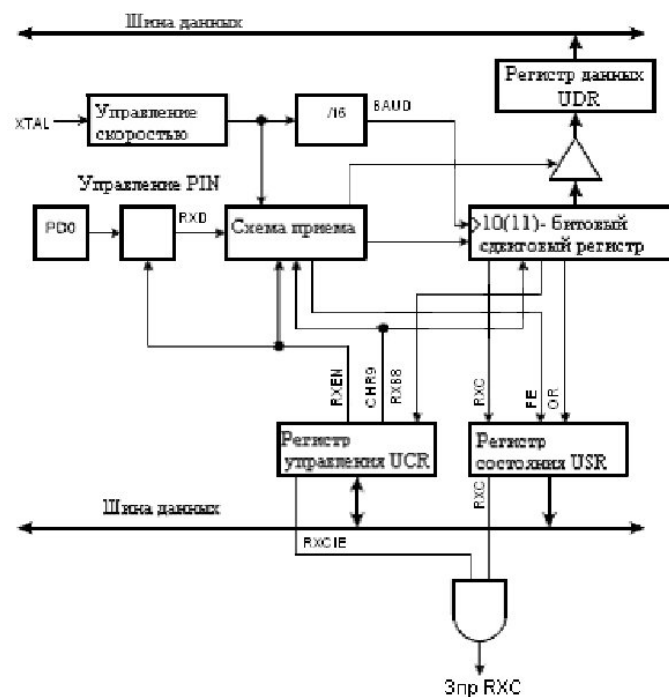


Рисунок 2 - структурная схема приемника модуля UART

Схемы алгоритмов передачи и приема приведены на рисунке 3.

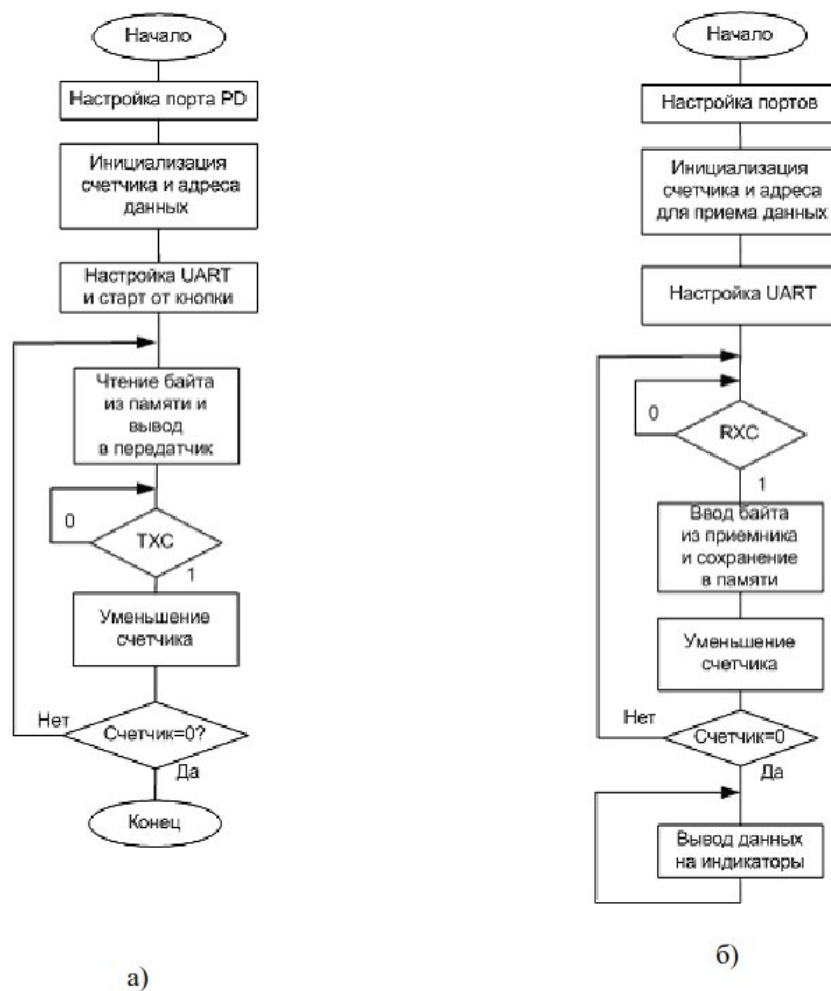


Рисунок 3 - схемы алгоритмов передачи и приема

В листингах 1 и 2 приведен исходный код программ передачи и приема соответственно.

#### Листинг 1 – программа передачи

```
;*****
;Программа 6.1 для МК ATx8515: 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 UBRR=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 UBRR (UBRR)
out UBRR,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 zl,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)

## Листинг 2 – программа приема

```
;*****
;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
```

Оценим скорость передачи. На рисунках 4 и 5 приведено состояние часов Stop Watch и регистров до передачи байта сообщения и после окончания передачи байта.

The screenshot displays the AVR Studio interface. On the left, the 'Registers' window shows the 'Stop Watch' value as 5.15 us. In the center, the assembly code is visible, with a red dot indicating the current execution point at the 'WAIT' label. On the right, the 'I/O View' window shows the status of various hardware registers, including PORTA, PORTB, PORTC, PORTD, DDRA, DDRB, DDRC, DDRD, PINA, PIND, PINC, PORTD, PORTE, SPI, TIMER\_COUNTER0, TIMER\_COUNTER1, USART, and WATCHDOG.

Name	Address	Value	Bits
ANALOG_COMP			
CPU			
EEPROM			
EXTERNAL_INTERRUPT			
PORTA			
PORTB			
PORTC			
PORTD			
DDRA	0x11 (0x31)	0x00	00000000
DDRB	0x10 (0x30)	0x02	00000010
DDRC	0x12 (0x32)	0x10	00010000
DDRD			
PINA			
PIND			
PINC			
PORTD			
PORTE			
SPI			
TIMER_COUNTER0			
TIMER_COUNTER1			
USART			
WATCHDOG			

Рисунок 4 – значение Stop Watch до передачи байта

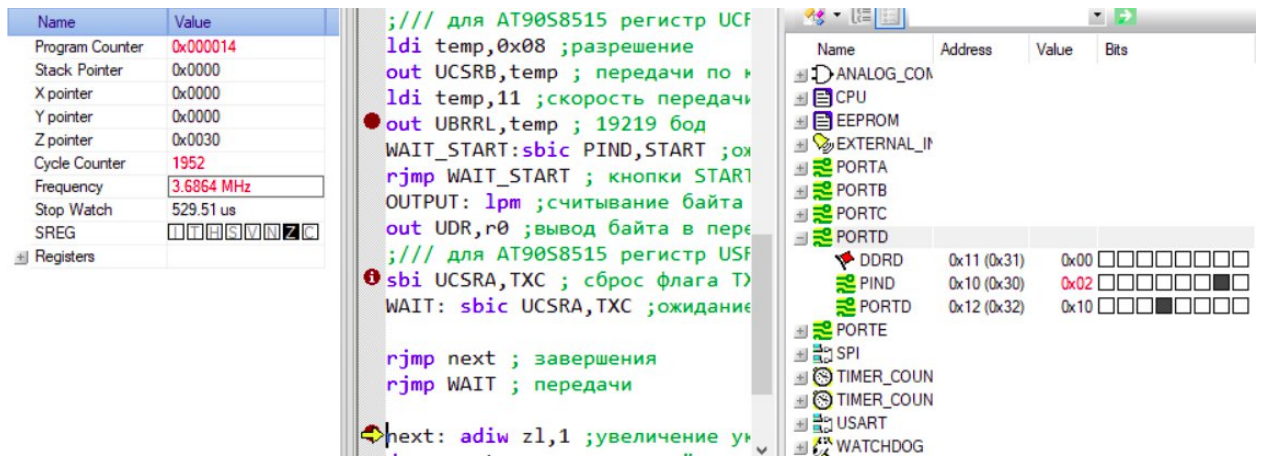


Рисунок 5 – значение Stop Watch после передачи байта

Отсюда  $V = 10 / (t_2 - t_1) = 10 / ((529.51 - 5.15) * 10^{-6}) = 19\,070$  бод. Данное значение близко к теоретическому значению в 19 200 Кбод.

Проверим программу приема. На рисунке 6 показан процесс ввода значения 0x68. На рисунках 7 и 8 – содержимое памяти и вывод значения на индикаторы соответственно.

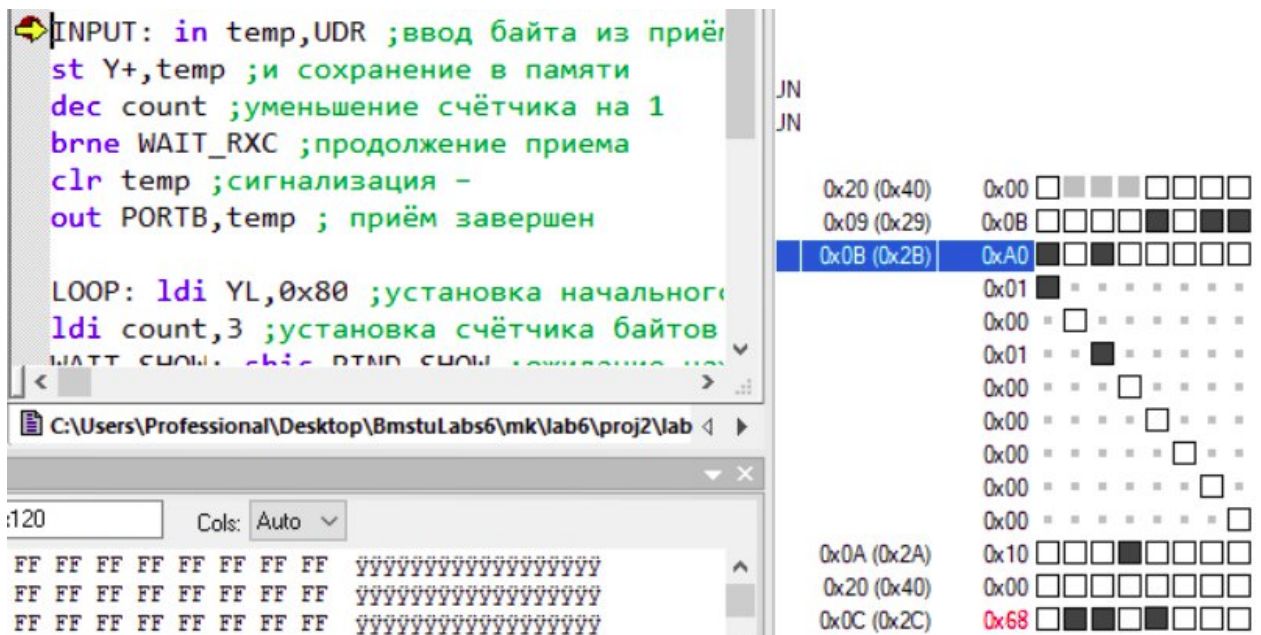


Рисунок 6 – ввод значения 0x68

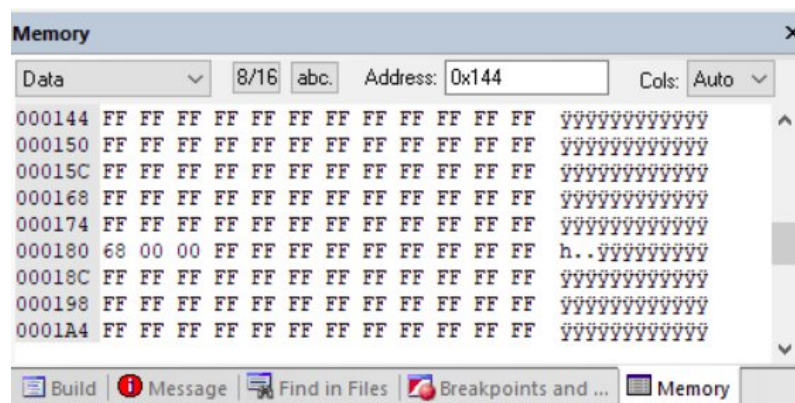


Рисунок 7 – содержимое памяти



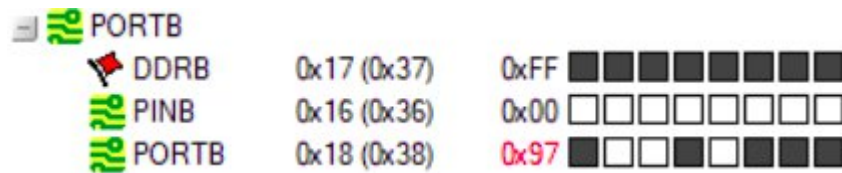


Рисунок 8 – вывод значения на индикаторы

Проверим корректность работы программ с помощью Proteus. На рисунках 9-11 приведены: схема в Proteus, временная диаграмма и содержимое памяти ведомого микроконтроллера соответственно.

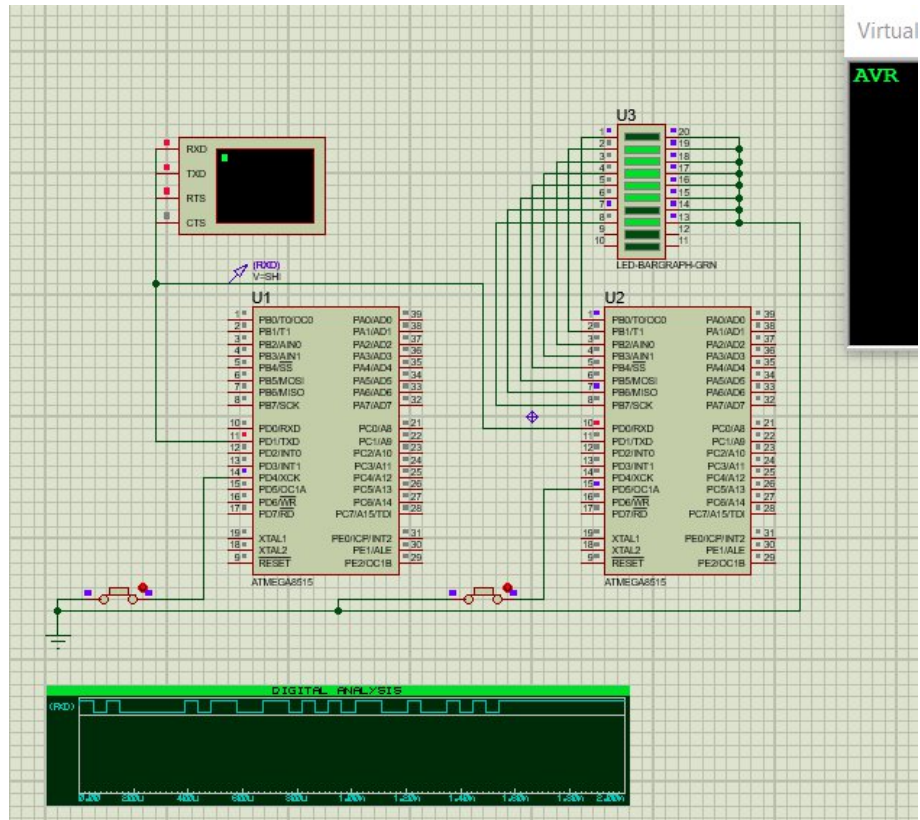


Рисунок 9 – схема в Proteus



Рисунок 10 – временная диаграмма



00000150	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	.....
00000160	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	.....
00000170	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	.....
00000180	41 56 52 00	00 00 00 00	00 00 00 00	00 00 00 00	AVR.....
00000190	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	.....
000001A0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	.....

Рисунок 11 – содержимое памяти ведомого микроконтроллера

## Задание 2

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

Изменим код программ передающего и принимающего микроконтроллера для передачи сообщения «1248abc».

Для передающего МК необходимо изменить:

```
ldi count,3 -> ldi count,7
text: .db 'A','V','R' -> text: .db '1','2','4','8','a','b',
'c'
```

Для принимающего МК необходимо изменить:

```
ldi count,3 -> ldi count,7
```

Содержимое виртуального терминала и памяти ведомого микроконтроллера представлены на рисунках 12-13 соответственно.

## Virtual Terminal

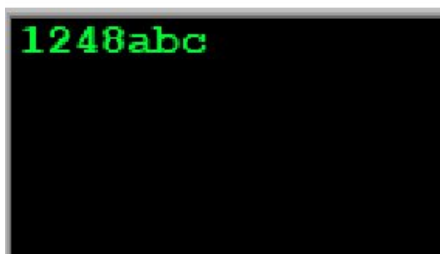


Рисунок 12 – окно виртуального терминала

00000170	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	.....
00000180	31 32 34 38	61 62 63 00	00 00 00 00	00 00 00 00	1248abc.....
00000190	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	.....

Рисунок 13 – содержимое памяти ведомого микроконтроллера

**Вывод:** в ходе лабораторной работы была изучена структура и принципы управления каналом UART (Universal Asynchronous Receiver-Transmitter). Кроме того, были получены навыки программирования передачи и приема данных по интерфейсу UART и моделирования работы канала в режиме обмена между двумя микроконтроллерами.