Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

Кафедра «Электронные вычислительные машины»

Отчет о лабораторной работе №1

**«Асинхронный приёмо-передатчик»**

по дисциплине

«ЭВМ и ПУ»

**Выполнили:**

**Проверил:**

Рязань 20

**Цель работы:** изучить принцип работы последовательного интерфейса UART, изучить порядок настройки и работы с интерфейсом на микроконтроллере Atmega328P.

**Практическая часть**

**Вариант 1**

**Пример 1.** Настройка интерфейса и передача одного байта данных. Пример отправки байта информации при нажатии на кнопку.

**Код программы для примера 1**

#include <avr/io.h>

#include <avr/interrupt.h>

ISR (INT0\_vect)

{

UDR0 = 0x3E;

}

int main(void)

{

EICRA = 0x03;

EIMSK = 0x01;

DDRD = 0x04;

PORTD |= 0x04;

UBRR0 = 103;

UCSR0B = (1<<TXEN0);

UCSR0C = (1<<UCSZ01)|(1<<UCSZ00);

sei();

while (1)

{

}

}

**Результат работы программы примера 1 (рисунок 1)**

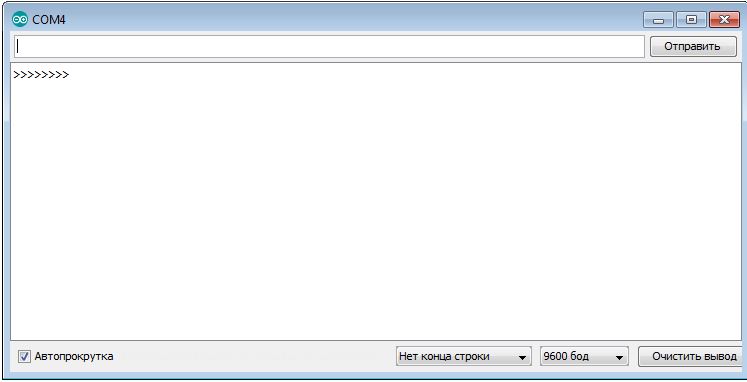
****

Рисунок 1 – Пример 1

**Задание 1.** Изменить передаваемый символ на «=». Выполнить сборку программы. Подключить логический анализатор к выводу PORTD0. Загрузить код в контроллер. Открыть программу Logic. Выполнить анализ сигнала с линии TX.

**Код программы для задания 1**

#include <avr/io.h>

#include <avr/interrupt.h>

ISR (INT0\_vect)

{

UDR0 = 0x3D;

}

int main(void)

{

EICRA = 0x03;

EIMSK = 0x01;

DDRD = 0x04;

PORTD |= 0x04;

UBRR0 = 103;

UCSR0B = (1<<TXEN0);

UCSR0C = (1<<UCSZ01)|(1<<UCSZ00);

sei();

while (1)

{

}

}

**Результат работы программы задания 1 (рисунки 2 - 3)**

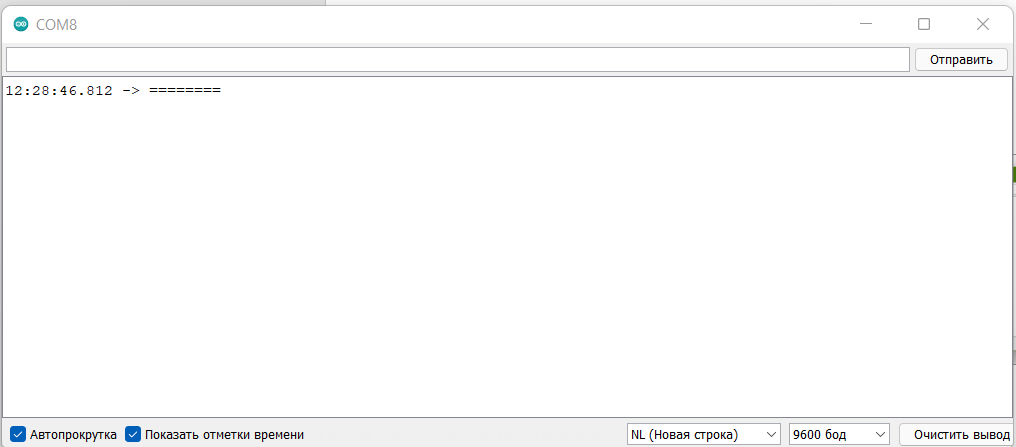


Рисунок 2 – Задание 1

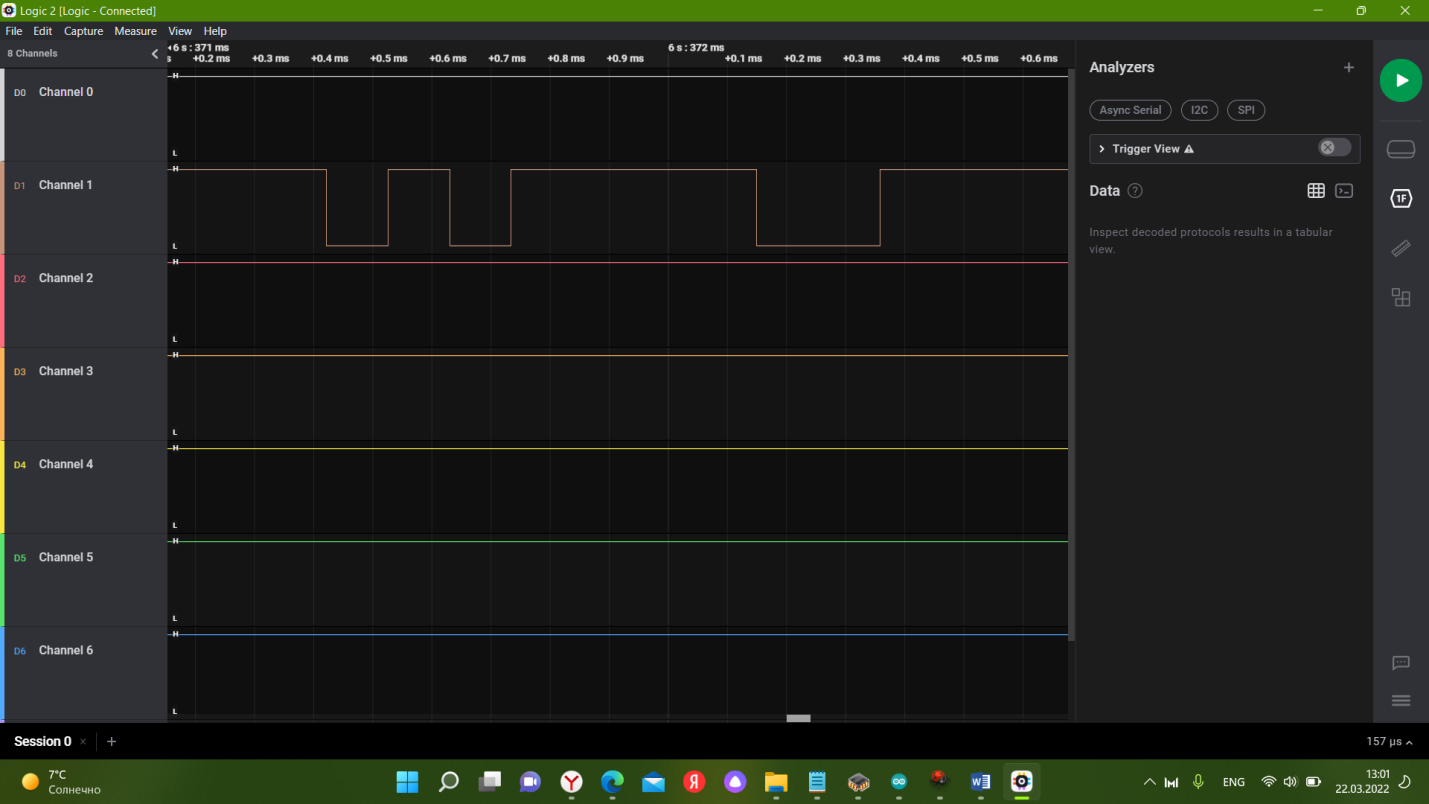


Рисунок 3 – Анализ сигнала с линии TX

**Пример 2.** Передача сообщений. В рассмотренных задачах выполняется передача по одному байту. Каждая передача выполняется по нажатию на клавишу. Пример передачи сообщения, записанного в ПЗУ микроконтроллера на языке Assembler. Программа постоянно выводит через UART одно и тоже сообщение.

**Код программы для примера 2**

.include "m328Pdef.inc"

.equ CLK=8000000

.equ BAUD=9600

.equ UBRR0\_value = (CLK/(BAUD\*16)) - 1

.org 0

jmp Reset

Reset:

;настройка UART

;установка частоты приёма/передачи

ldi r16, high(UBRR0\_value)

sts UBRR0H, r16

ldi r16, low(UBRR0\_value)

sts UBRR0L, r16

;разрешение передачи

ldi r16,(1<<< UCSZ00)|(1<< UCSZ01)

sts UCSR0C,R16

;разрешение всех прерываний

sei

main:

;определение адреса сообщения в ПЗУ

ldi zl,low(Msg<<1)

ldi zh, high(Msg<<1)

;количество байт в сообщении

ldi r18,0x17

send: ;отправка сообщения

lpm r16, z+

sts UDR0,r16

subi r18,0x01

repeat: ;проверка окончания сообщения

lds r17, UCSR0A

bst r17, 5

brtc repeat

cpi r18,0x00

breq main

jmp send

.cseg ;размещение сообщения в ПЗУ

Msg: .db “Atmega328P:ReadyToWork”, ‘\n’

**Результат работы программы примера 2 (рисунок 4)**

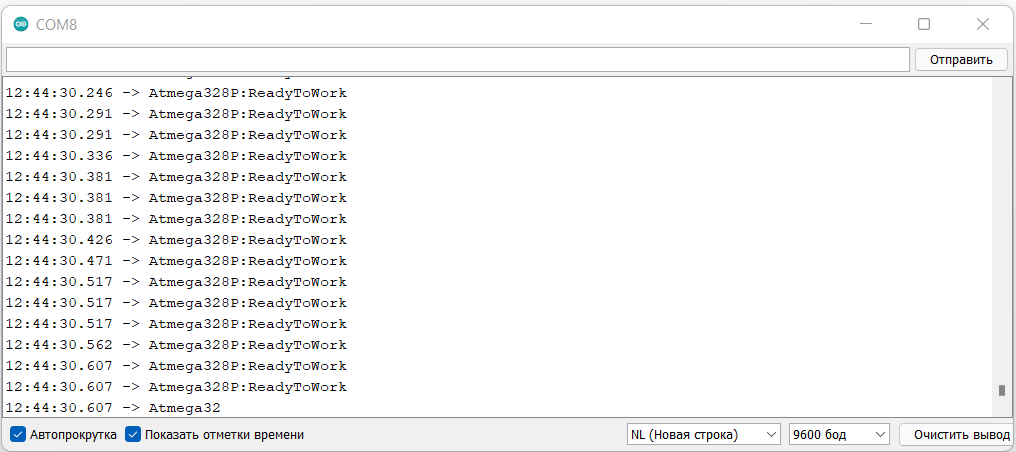


Рисунок 4 – Пример 2

**Задание 2.** Разработать аналогичную примеру 2 программу на языке C.

**Код программы для задания 2**

#include <avr/io.h>

#include <avr/interrupt.h>

int i = 0;

char Msg[23]={'A', 't', 'm', 'e', 'g', 'a', '3', '2', '8', 'P', ':', 'R', 'e', 'a', 'd', 'y', 'T', 'o', 'W', 'o', 'r', 'k', '\n'};

ISR (USART\_TX\_vect)

{

if (i <= 22)

{

i++;

UDR0 = Msg[i];

}

else

{

i = 0;

UDR0 = Msg[i];

}

}

int main(void)

{

UBRR0 = 103;

UCSR0B = (1<<TXCIE0)|(1<<TXEN0);

UCSR0C = (1<<UCSZ01)|(1<<UCSZ00);

if (i == 0)

{

UDR0 = Msg[0];

}

sei();

while(1)

{

}

}

**Результат работы программы задания 2 (рисунок 5)**

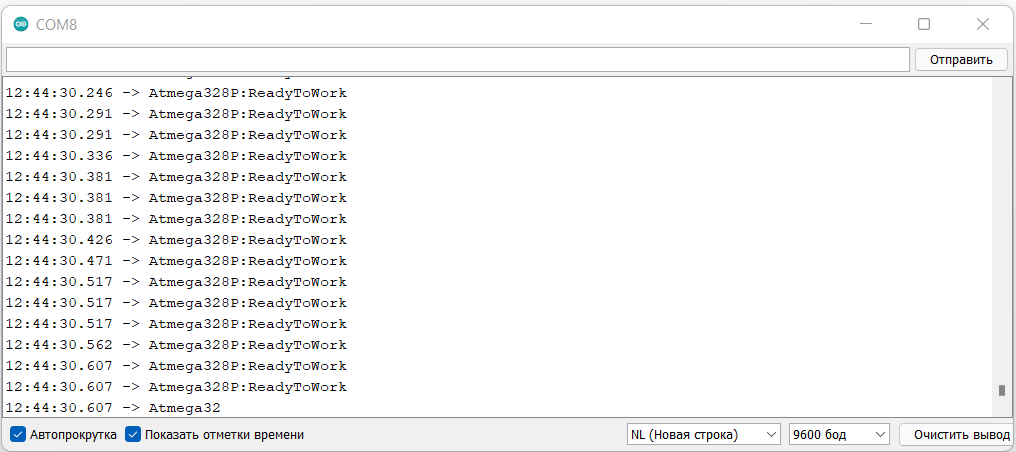


Рисунок 5 – Задание 2

**Задание 3.** Прием информации: исправьте код примера таким образом, чтобы плата реагировала на принятый байт (А) следующим образом: если получен символ согласно номеру варианта, светодиод L (PortB5) должен зажечься, в противоположном случае – погаснуть.

**Код программы для задания 3**

#include <avr/io.h>

#include <avr/interrupt.h>

int zn;

ISR (USART\_RX\_vect)

{

zn = UDR0;

if (zn == 'A')

{

PORTB = 0x20;

}

else

{

PORTB = 0x00;

}

}

int main(void)

{

DDRB = 0xFF;

UBRR0 = 103;

UCSR0B = (1<<RXEN0)|(1<<RXCIE0);

UCSR0C = (1<<UCSZ01)|(1<<UCSZ00);

sei();

while (1)

{

}

}

**Задание 4.** Обмен пакетами данных. Разработайте программу, которая получает 4 числа в формате !a,b,c,d. (где a,b,c и d – целые числа, «!» – начало обмена, «.» – завершение обмена), и в порядке b,a,d,c передает обратно.

**Код программы для задания 4**

#include <avr/io.h>

#include <avr/interrupt.h>

int i = 0;

int zn, fl;

int Mess[4];

ISR (USART\_RX\_vect)

{

zn = UDR0;

if (zn == '!')

{

fl = 1;

i++;

}

if ((fl == 1) && (zn != '.') && (i == 1))

{

Mess[i] = zn;

i++;

}

if ((fl == 1) && (zn != '.') && (i == 2))

{

Mess[i] = zn;

i++;

}

if ((fl == 1) && (zn != '.') && (i == 3))

{

Mess[i] = zn;

i++;

}

if ((fl == 1) && (zn != '.') && (i == 4))

{

Mess[i] = zn;

i=0;

}

if (zn == '.')

{

UDR0 = Mess[2];

i++;

}

}

ISR (USART\_TX\_vect)

{

if ((i!=4) && (i==1))

{

UDR0 = Mess[1];

i++;

}

if ((i!=4) && (i==2))

{

UDR0 = Mess[4];

i++;

}

if ((i!=4) && (i==3)){

UDR0 = Mess[3];

i=0;

}

}

int main(void)

{

UBRR0 = 103;

UCSR0B = (1<<TXCIE0)|(1<<TXEN0)|(1<<RXCIE0)|(1<<RXEN0);

UCSR0C = (1<<UCSZ01)|(1<<UCSZ00);

sei();

while(1)

{

}

}

**Результат работы программы задания 4 (рисунок 6)**

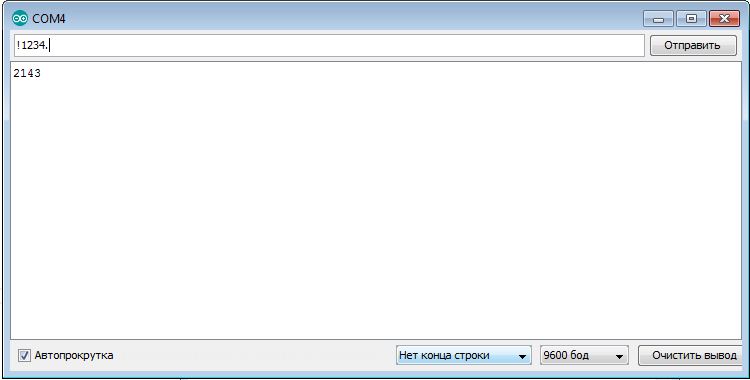
****

Рисунок 6 – Задание 4

**Вывод.** В ходе работы над данным практическим заданием изучили принцип работы последовательного интерфейса UART, изучили порядок настройки и работы с интерфейсом на микроконтроллере Atmega328P.