

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ОБРАЗОВАНИЯ «БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Лабораторная работа №2  
дисциплина «ЭВМ и периферийные устройства»  
по теме «Изучение принципов организации обмена данными по  
последовательному интерфейсу USB между микроконтроллером MSP430F1611  
и ПЭВМ»

Выполнил: студент группы ВТ-31

Проверил:

Макаров Д.С.

Шамраев А.А.

Белгород 2020

# Лабораторная работа №2

## «Изучение принципов организации обмена данными по последовательному интерфейсу USB между микроконтроллером MSP430F1611 и ПЭВМ»

**Цель работы:** Изучить возможности сопряжения лабораторного стенда на базе микроконтроллера MSP430F1611 и ПЭВМ с помощью последовательного интерфейса USB, принципы программного управления двунаправленным обменом данных по последовательному интерфейсу USB.

### Вариант 6

**Задание:** Разработать программу передачи 50 чисел из ПЭВМ в микроконтроллер по интерфейсу USB в соответствии с протоколом:

- модуль UASRT0
- скорость 14400 бит/с
- режим обмена - асинхронный
- 7 битов без бита четности

### Порядок выполнения задания:

- включить лабораторный макет.
- запустить Code Composer IDE.
- создать пустой проект.
- создать файл ресурса для кода программы и подключить его к проекту.
- выполнить компиляцию исходного модуля программы и устранить ошибки, полученные на данном этапе.
- проверить работоспособность программы и показать результаты работы преподавателю.

### Ход работы

### Схема стенда

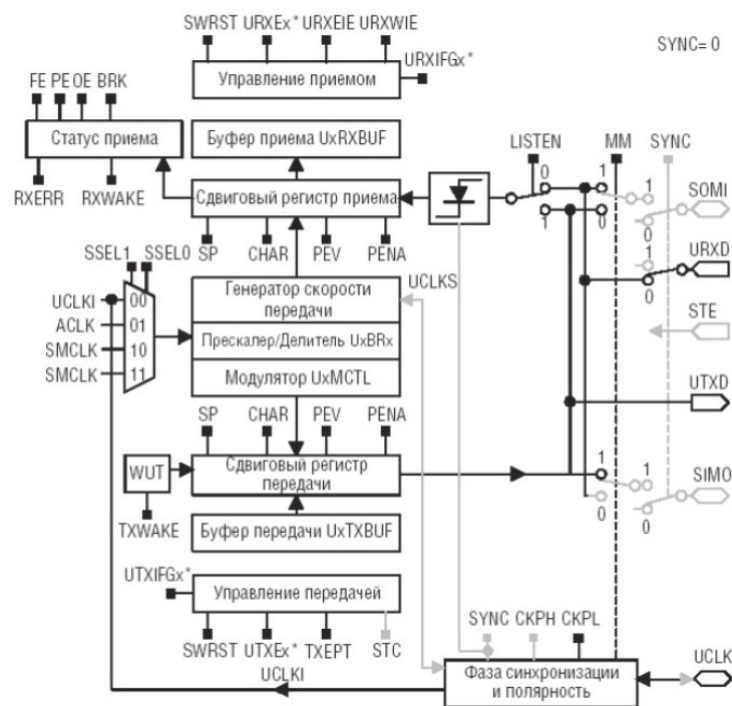


Рис. 1: Схема UASRT в режиме UART



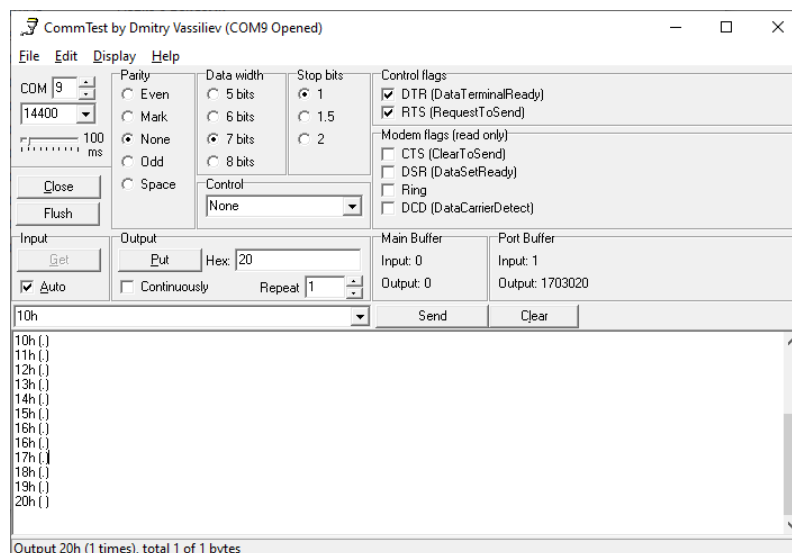


Рис. 4: Полученные данные

**Вывод:** Я изучил принципы программного управления двунаправленным обменом данных по USB, между микроконтроллером семейства MSP и ПК.

# Приложение

## Содержимое файла main.c

```
#include <msp430.h>
#include "stdio.h"
#include "system_define.h"
#include "system_variable.h"
#include "function_prototype.h"
#include "main.h"

#define COUNT_OF_NUMBERS 50
// speed = 0 - 38400, 1 - 57600, 2 - 115200 - скорость обмена
// databits = 7, 8 - длина символа
// stopbits = 1, 2 - кол-во передаваемых стоповых бит
// parity = 0 - без контроля четности, 1 - контроль четности, нечетный, 2 - четный
// iface = 0 - USB, 1 - оптика
void main(void) {
    WDTCTL = WDTPW + WDTHOLD;
    Init_System_Clock();
    UART_init(7,1,0,0);
    UART_message("Ready to transmit\n");

    int i;
    char numbers[COUNT_OF_NUMBERS];
    for(i = 0; i < COUNT_OF_NUMBERS; i++){
        numbers[i] = UART_getbyte();
        UART_sendbyte(numbers[i]);
    }
    while(1);
}
```

## Содержимое файла sysfunc.c

```
// System functions

#include <msp430.h>
#include "sysfunc.h"

// инициализация портов системы
void Init_System()
{
    P1DIR |= (nSS + nWR_nRST + MCU_SEL_0 + MCU_SEL_1); // установка направления портов на
    ↪ вывод
    DB_DIR = 0x00; // шина данных настроена на ввод
}

// инициализация системы тактирования
void Init_System_Clock()
{
    volatile byte i;
    BCCTL1 &= ~XT2OFF; // включение осциллятора XT2
    // MCLK = XT2, SMCLK = XT2
    // ожидание запуска кварца
    do
    {
        IFG1 &= ~OFIFG; // Clear OSCFault flag
        for (i = 0xFF; i > 0; i--); // Time for flag to set
    }
}
```

```

    }
    while ((IFG1 & OFIFG));           // OSCFault flag still set?
    BCSTL2 |= SELM_2 | SELS;         // установка внешнего модуля тактирования
}

// 2do: сделать точную задержку
void wait_1ms(word cnt)
{
    for (wait_i = 0; wait_i < cnt; wait_i++)
        for (wait_j = 0; wait_j < 1000; wait_j++);
}

void wait_1mks(word cnt)
{
    for (wait_i = 0; wait_i < cnt; wait_i++);
}

```

## Содержимое файла uart.c

```

// UART functions
#include "function_prototype.h"
#include "sysfunc.h"
#include "uart.h"

// Инициализация режима UART
// speed = 0 - 38400, 1 - 57600, 2 - 115200 - скорость обмена
// databits = 7, 8 - длина символа
// stopbits = 1, 2 - кол-во передаваемых стоповых бит
// parity = 0 - без контроля четности, 1 - контроль четности, нечетный, 2 - четный
// iface = 0 - USB, 1 - оптика
void UART_init(byte databits, byte stopbits, byte parity, byte iface)
{
    P3SEL |= BIT6 | BIT7;           // выбор функции USART1
    U1CTL = 0;                       // инициализация состояния USART
    ME2 |= UTXE1 + URXE1;           // включить приемник и передатчик USART1

    if (databits == 7) U1CTL &= ~CHAR; // 7-разрядная длина символа
    if (databits == 8) U1CTL |= CHAR;  // 8-разрядная длина символа
    if (stopbits == 1) U1CTL &= ~SPB;  // 1 стоповый бит
    if (stopbits == 2) U1CTL |= SPB;   // 1 стоповых бита
    if (parity == 0) U1CTL &= ~PENA;    // контроль четности отключен
    if (parity == 1) U1CTL = (U1CTL & ~PEV) | PENA; // контроль четности, нечетный
    if (parity == 2) U1CTL |= PENA | PEV; // контроль четности, четный

    P5DIR |= BIT0;                  // переключение мультиплексора на USB/оптику
    if (iface == 0)
        P5OUT |= BIT0;
    if (iface == 1)
        P5OUT &= ~BIT0;
    U1TCTL |= SSEL1;

    //U1BR = Тактовая частота/требуемая скорость
    //Младший разряд
    U1BR0 = 0x2B;
    //Старший разряд
    U1BR1 = 0x02;
    //Дробная часть
    U1MCTL = 0x37;
}

```

```

// отключение режима UART
void UART_off()
{
    P3SEL |= BIT6 | BIT7;           // выбор функции USART1
    ME2 &= ~(UTXE1 + URXE1);        // выключить приемник и передатчик USART1
    U1CTL = SWRST;                   // отключение USART1
}

// вывод строки символов (символ с кодом 0 - конец строки)
void UART_message(char * buf)
{
    word i=0;
    while (buf[i])
        UART_sendbyte(buf[i++]);    // передача символа
}

// передача байта
void UART_sendbyte(char byte)
{
    while (!(IFG2 & UTXIFG1));       // проверка готовности буфера передачи USART1
    U1TXBUF = byte;                  // передача байта
}

// получение байта
char UART_getbyte()
{
    while (!(IFG2 & URXIFG1));       // проверка готовности буфера приема USART1
    return U1RXBUF;                  // возврат полученного байта
}

```