**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)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

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: ст. группы ВТ-31  
Новожен Н.В

Проверил: Шамраев А,А

**Белгород 2020**

**Цель работы** : изучить принципы функционирования и возможности программного управления цифровым индикатором, разработать алгоритм и программу для вывода информации на цифровой индикатор.

**Ход работы**

– характеристики лабораторной вычислительной системы;

– исходный модуль разработанной программы;

– анализ полученных результатов и краткие выводы по работе, в которых необходимо отразить особенности использования встроенных в микроконтроллер модулей USART при реализации обмена данными между лабораторным стендом и ПЭВМ

**Вариант-7**

Разработать программу передачи 200 чисел (от 0 до 199) из микроконтроллера MSP430F1611 в ПЭВМ по интерфейсу USB в соответствие с протоколом:

модуль USART0, скорость обмена данными 38400 бит/с, режим обмена асинхронный, 8 битов данных без бита четности.

**void** **main**(**void**) {

WDTCTL=WDTPW+WDTHOLD;

Init\_System();

Init\_System\_Clock();

UART\_init(0,8,1,0,0);

**char** i=0;

**while**(1){

i=0;

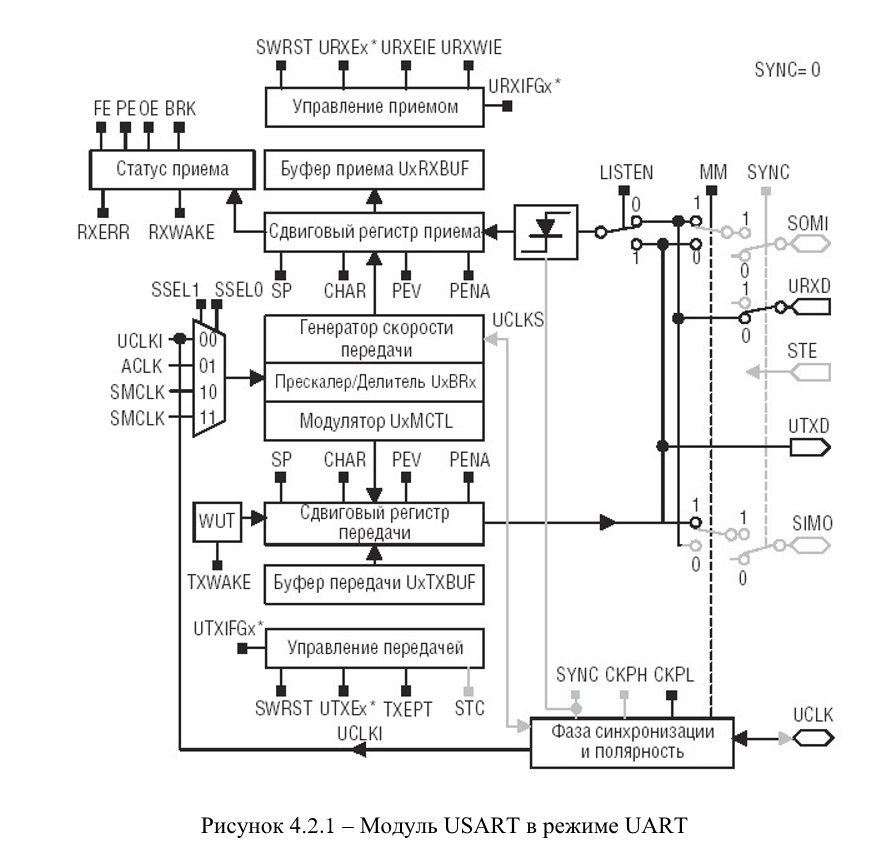
**for**(;i<200;i++)

UART\_sendbyte(i);

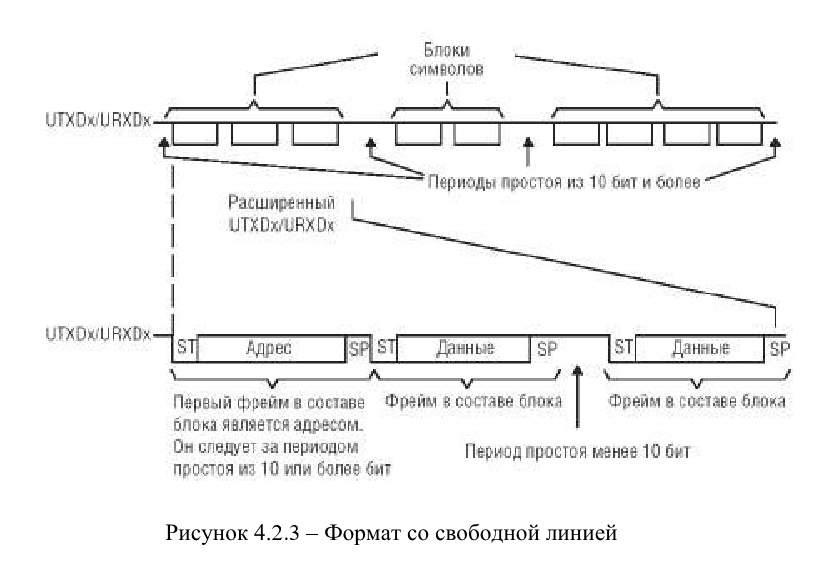
wait\_1ms(1000);

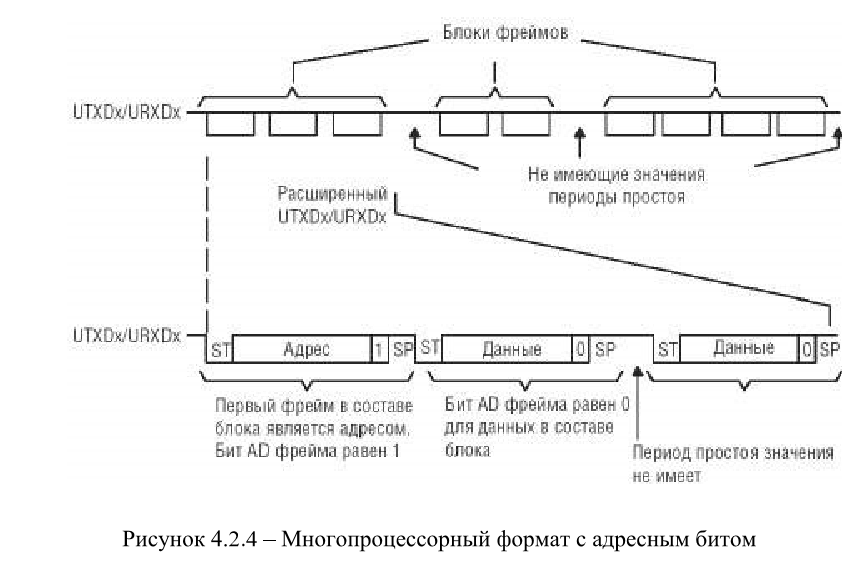
}

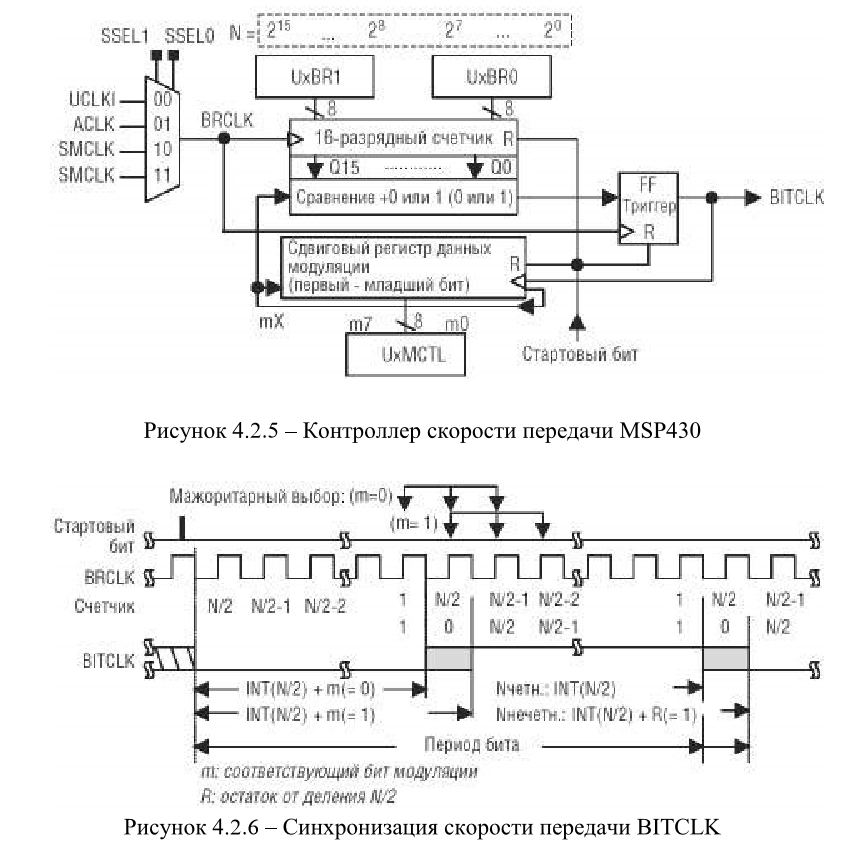
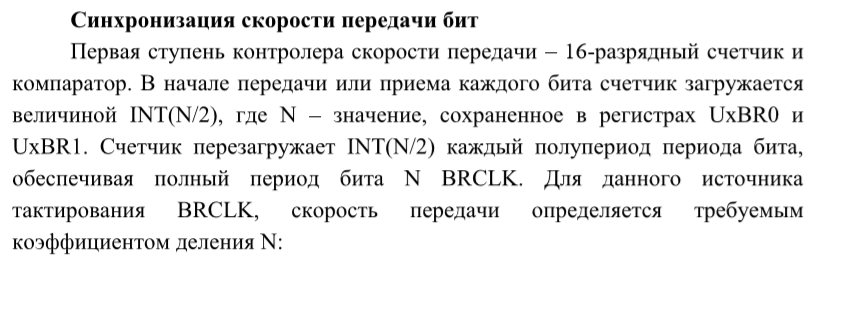
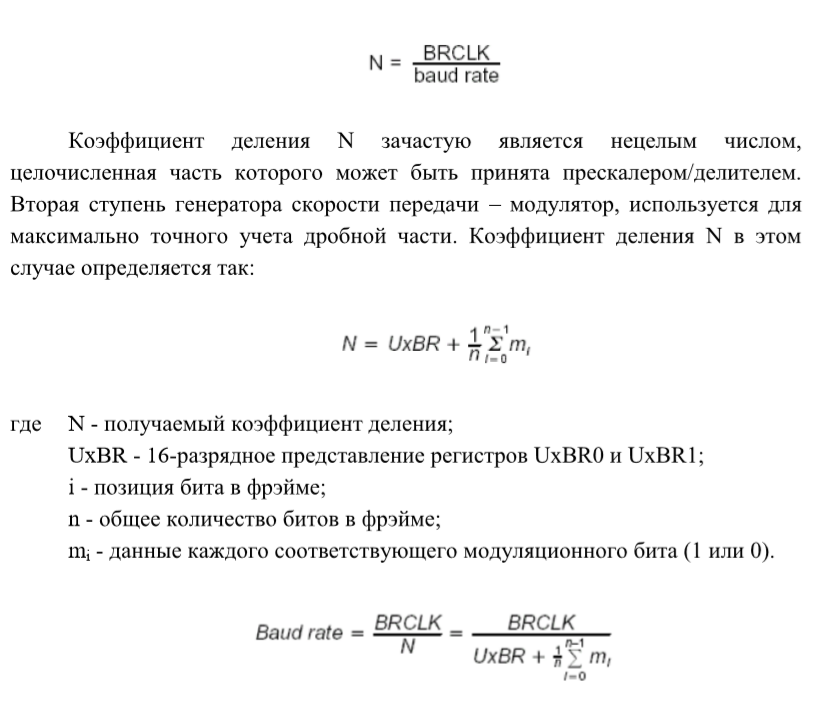
}

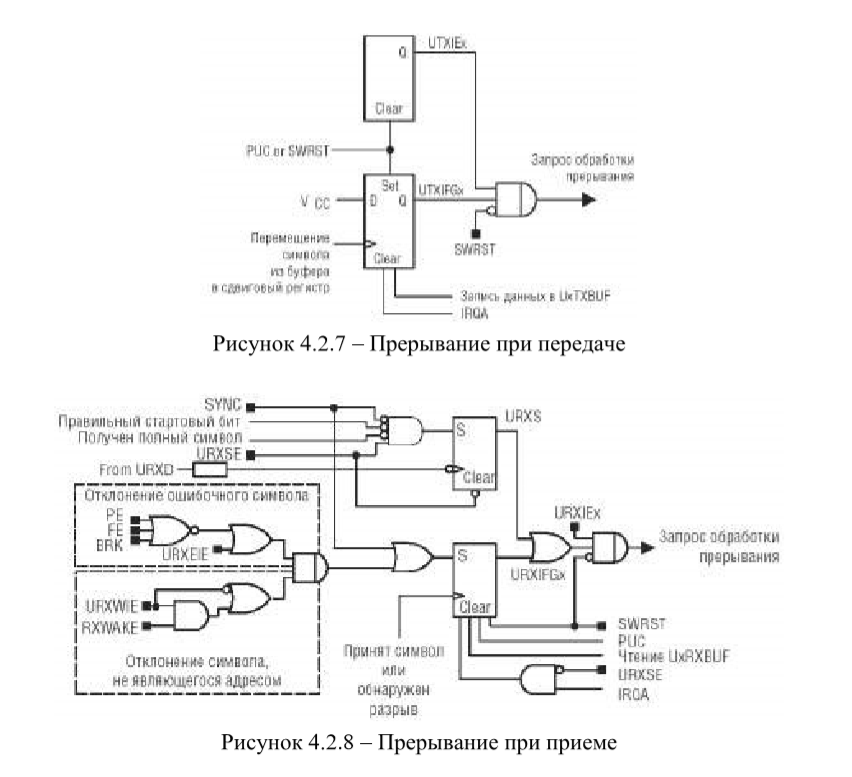
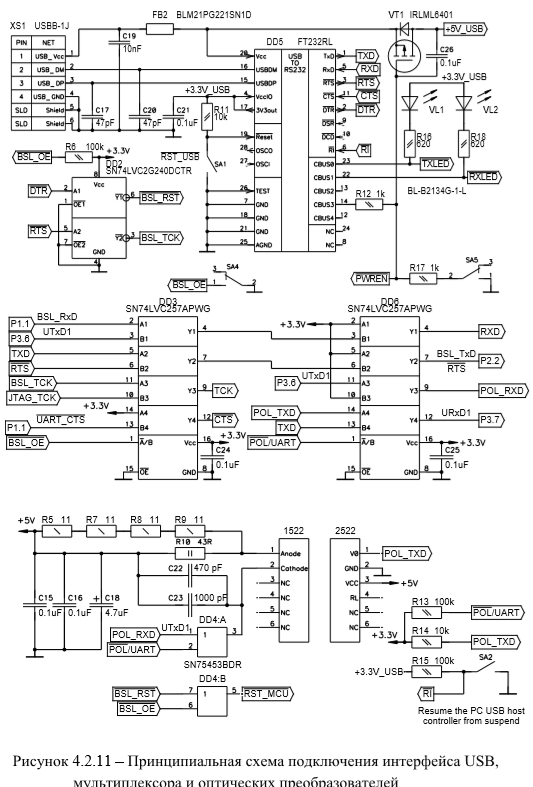


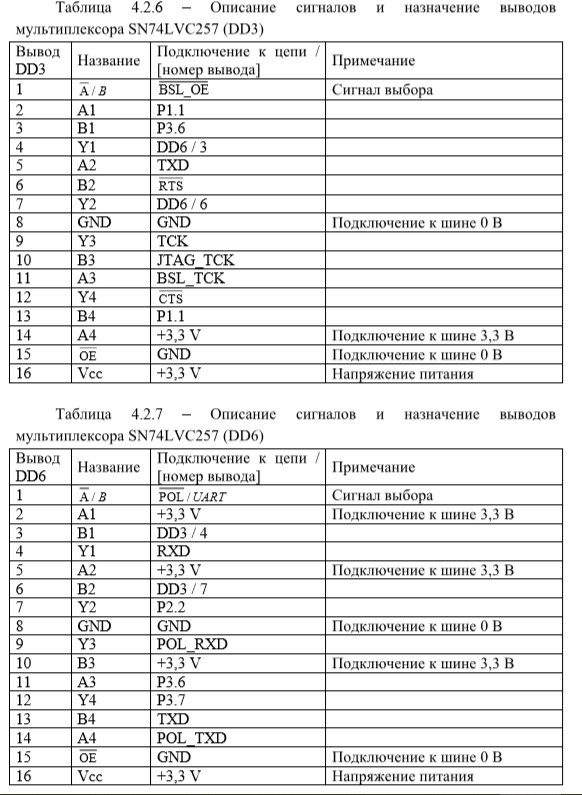








**Порядок выполнения задания:**

– включить лабораторный макет.

– запустить компилятор IAR Embedded Workbench.

– создать пустой проект.

– создать файл ресурса для кода программы и подключить его к проекту.

– ввести код исходного модуля программы обмена данными между микроконтроллером MSP430F1611 с ПЭВМ по интерфейсу USB в соответствие с индивидуальным заданием, приведенным в таблице 4.2.8.

– выполнить компиляцию исходного модуля программы и устранить ошибки, полученные на данном этапе.

– настроить параметры программатора.

– проверить правильность подключения интерфейсного USB кабеля к разъемам лабораторного макета и ПЭВМ.

– запустить на ПЭВМ программу Terminal, установить необходимые параметры протокола обмена данными, выбрать номер последовательного порта (СОМx), соответствующего виртуальному COM-порту, и нажать на кнопку Connect в верхнем левом углу рабочего окна программы.

– создать загрузочный модуль программы и выполнить программирование микроконтроллера. – проверить работоспособность загруженной в микроконтроллер программы и показать результаты работы преподавателю. В случае некорректной работы разработанной программы, выполнить аппаратный сброс микроконтроллера, провести отладку исходного модуля программы и заново проверить функционирование программы.

**Вариант-7**

**void** **main**(**void**) {

WDTCTL=WDTPW+WDTHOLD;

Init\_System();

Init\_System\_Clock();

UART\_init(0,8,1,0,0);

**char** i=0;

**while**(1){

i=0;

**for**(;i<200;i++)

UART\_sendbyte(i);

wait\_1ms(1000);

}

}

**void** **UART\_init**(byte speed, 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 длинна символа

**if** (databits == 8) U1CTL |= CHAR; // 8-разрядная длинна 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; // BRCLK = SMCLK

U1BR0 = 69; // 8Mhz / 115200 = 69.44 (по-умолчанию)

**if** (speed == 0) U1BR0 = 208; // 8МГц / 38400 = 208.33

**if** (speed == 1) U1BR0 = 139; // 8МГц / 57600 = 138,89

U1BR1 = 0x00;

U1MCTL = 0x2C; // модуляция

}

// передача байта

**void** **UART\_sendbyte**(**char** byte)

{

**while** (!(IFG2 & UTXIFG1)); // проверка готовности буфера передачи USART1

U1TXBUF = byte; // передача байта

}

// 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;

BCSCTL1 &= ~XT2OFF; // включение осцилятора XT2

// MCLK = XT2, SMCLK = XT2

**do** // ожидание запуска кварца

{

IFG1 &= ~OFIFG; // Clear OSCFault flag

**for** (i = 0xFF; i > 0; i--); // Time for flag to set

}

**while** ((IFG1 & OFIFG)); // OSCFault flag still set?

BCSCTL2 |= SELM\_2 | SELS; // установка внешнего модуля тактирования

}

// 2do: сделать точную задержку

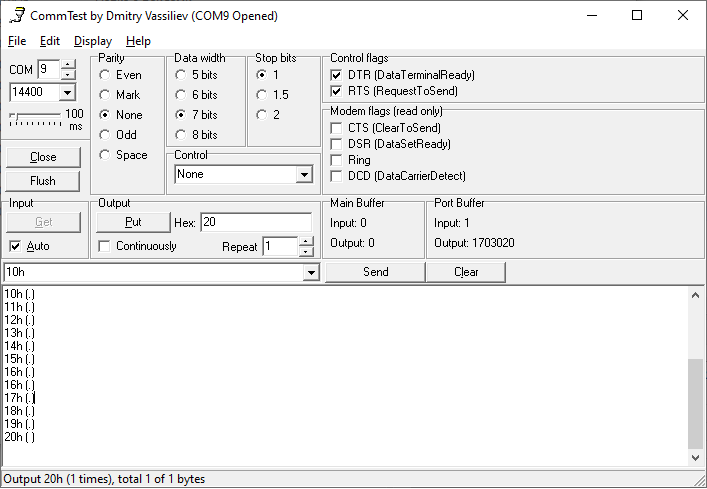
**void** **wait\_1ms**(word cnt)

{

**for** (wait\_i = 0; wait\_i < cnt; wait\_i++)

**for** (wait\_j = 0; wait\_j < 1000; wait\_j++);

}



**Вывод:** мы изучение принципы организации обмена данными по последовательному интерфейсу usb между микроконтроллером msp430f1611 и пэвм

1. Поясните принципы передачи информации по последовательным и параллельным интерфейсам.

2. Назовите современные универсальные интерфейсы и приведите их основные характеристики.

3. Поясните принципы обмена данными по интерфейсу USB.

4. Какие регистры используются для настройки параметров передачи данных с помощью встроенного в микроконтроллер MSP430F1611 блока USART?

5. Какие сигналы прерываний могут генерироваться блоком USART?

6. Поясните формат кадра при обмене данными в форматах со свободной линией и адресным битом.

