**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕНЫЙ 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

Тема: «Изучение принципов организации обмена данными по последовательному интерфейсу usb между микроконтроллером MSP430f1611 и ПЭВМ»

Выполнил студент группы ВТ - 31

Проверил: Шамраев А.А.

Белгород 2019

**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.

**Задание**: Разработать программу передачи 50 чисел (от 20 до 69) 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.

**Порядок выполнения работы**:

– 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.

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

– настроить параметры программатора.

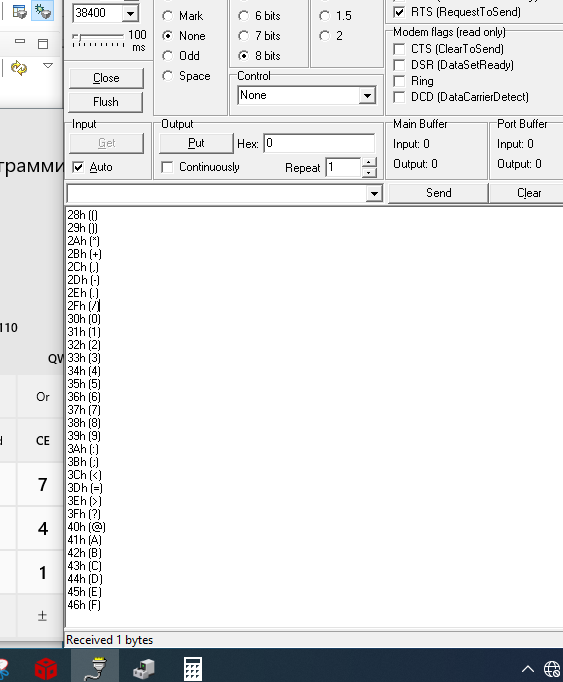
– проверить правильность подключения интерфейсного USB кабеля к разъемам лабораторного макета и ПЭВМ.

– 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.

**Демонстрация работы программы:**



**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"

void main(void) {

WDTCTL = WDTPW + WDTHOLD;

Init\_System\_Clock();

Init\_System();

char i, j = 20;

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

while(1)

{

for(i = 0; i < 50; i++)

{

UART\_sendbyte(j);

j++;

}

UART\_off();

}

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;

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++);

}

**void** **wait\_1mks**(word cnt)

{

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

}

**Содержимое файла uart.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;

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++);

}

**void** **wait\_1mks**(word cnt)

{

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

}

**Контрольные вопросы и задания**

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.