**Задание к лабораторной работе №3**

**Задание**

1. Выполните проверку работы программы в симуляторе AVR Studio. Обратите особое внимание проверке в симуляторе подпрограмм обработки прерываний Tx Complete, Data register Empty, RX Complete;
2. Выполните: программирование стенда EasyAVR5.Запустите программу AVRStend на ПК. Выполните передачу команды с ПК и прием ответа от стенда EasyAVR5. Проверьте правильность принятых и переданных данных на семисегментных индикаторах стенда EasyAVR5.
3. Доработайте программу сократив необоснованно большую паузу между нажатиями кнопки «Просмотр».
4. Проведите эксперимент по изменению и передаче команд арифметических операций и правильности их приема стендом EasyAVR5.
5. Измените скорость передачи в программе МК и программе AVRStend, проверьте и добейтесь правильной работы изменной программы.
6. Добавьте бит паритета в программах МК и AVRStend и проверьте и добейтесь правильной работы изменной программы.

**Этапы работы**

3.3 Проверьте правильность работы программы в симуляторе, для этого нажмите кнопку Assemble and run (ctrl+F7).

Добавьте в окно Watch все перечисленные ранее регистры, а также регистры XL, XH, YL, YX, ZH и ZL. Настройте столбец Value окна Watch так, чтобы он отображал значения регистров в шестнадцатиричном формате.

Откройте окно Memory (View – Memory) и закрепите его в рабочем пространстве. В качестве контролируемых данных выберите Data в выпадающем списке окна Memory. Нажимая кнопку abs в окне Memory, добейтесь, чтобы в одной строке отображалось восемь бит памяти данных.

Используя клавишу F11 (пошаговое выполнение программы) дойдите до метки *Start*. При этом происходит инициализация используемой периферии, сбрасываются буферы в памяти данных, присваивается начальное значение регистрам. Наблюдайте в окне I/O View изменение состояния регистров при инициализации. Выводы PD4, PD7 МК используются и конфигурируются как входы. Логическое состояние на них задается при помощи регистра *PIND*. Неактивное состояние кнопки соответствует логической «1», поэтому необходимо установить соответствующие биты регистра PIND в единицу.

Начиная с метки *Start* начинается основная программа. На первой ее строчке вызывается подпрограмма динамической индикации *Display*. Нажмите несколько раз кнопку F11, наблюдая за работой программы. Вы можете увидеть, как постепенно уменьшается значение регистра *Disp\_Count*, который отвечает за формирование времени активности семисегментного индикатора.

Симулируйте прием одного кадра данных приемников модуля USART. Для этого выставите флаг прерывания по завершению приема RXC в регистре *UCSRA* в окне *View* (седьмой бит). Нажмите один раз кнопку F11, Вы перейдете на соответствующий вектор прерывания. Нажмите несколько раз кнопку F11 и дойдите до строки с меткой *rd\_UDR*, на строке с данной меткой происходит чтение регистра принятых данных *UDR*. Чтобы прочитать из этого регистра данные, отличные от ноля, необходимо перед циклом чтения записать в регистр требуемые данные. В окне View введите значение 0х01 в регистр *UDR* и нажмите *Enter*. **Будьте внимательны, при симулировании данные в регистре UDR хранятся только один такт, на следующий такт они сбрасываются.** Нажмите кнопку F11 несколько раз, чтобы выйти из подпрограммы обработки прерывания. Наблюдайте при этом за изменением регистров в окне *View* и буферов в памяти данных в окне *Memory*.

Симулируйте прием еще трех кадров данных по приведенной выше методике. Последовательно записывайте данные: 0хАА, 0х55, 0хFF. Данная последовательность означает, что будет выполнять операция суммирования (0х01) чисел 170 (0хАА) и 85 (0х55).

После выполнение перечисленных выше действий МК готов к исполнению арифметических операций над принятыми данными. Нажмите кнопку F11 несколько раз до перехода на строчку с меткой *Receive*. С этой строки начинается исполнение ряда подпрограмм для обработки данных: *Wr\_D\_rec*, *arifm\_op*, *pre\_date*, *Wr\_D\_tr*. Нажимая кнопку F11 внимательно просимулируйте работу каждой из подпрограмм. После выполнения всех перечисленных подпрограмм в окне *Memory* должны обновиться данные во всех буферах.

Далее Вы попадете на строку с меткой *Wait*. Начиная с этой строки, находится группа команд, проверяющих условие нажатия кнопок «Просмотр» (меняет отображаемый на семисегментных индикаторах байт) и «Ответ» (инициирует передачу информации с выхода модуля USART). Симулируйте нажатие кнопки «Просмотр» переводом соответствующего бита регистра *PIND* в ноль. Нажмите несколько раз кнопку F11 до перехода в подпрограмму *ch\_pos\_D*. Полностью просимулируйте работу этой подпрограммы, следя за изменением регистров в окне *View*. По завершении работы подпрограммы значение регистров *Number,* *Hundreds*, *Tens* и *Ones* должно измениться. Симулируйте отпускание кнопки «Просмотр» переводом соответствующего бита регистра *PIND* в единицу.

Симулируйте нажатие кнопки «Ответ» переводом соответствующего бита регистра *PIND* в ноль. Нажмите клавишу F11 несколько раз до перехода к строке с меткой *Trans*. В данной строке разрешается прерывание по опустошению буфера передачи модуля USART. Флаг этого прерывания будет уже установленным, поэтому нажмите кнопку F11 несколько раз до перехода к подпрограмме обработки прерывания по опустошению буфера передачи. В данной подпрограмме осуществляется запись первого байта данных в регистр *UDR*, запрещается прерывание по опустошению буфера и разрешается прерывание по окончанию передачи. Нажмите кнопку F11 несколько раз, чтобы выйти из данной подпрограммы.

Программа перейдет на строку с меткой *wait\_tr* и будет находиться там до окончания передачи данных. Передача следующего кадра будет осуществлена после завершения передачи текущего кадра. Чтобы симулировать работу прерываний Tx Complete необходимо выставить точку останова на строке rjmp TRANdate в описании векторов прерываний. Нажмите кнопку F5, наблюдайте переход к подпрограмме обработки прерывания по завершению передачи. Нажимая кнопку F11, пройдите ее полностью, при этом обращайте внимание на изменение регистров в окне *Watch* и регистра UDR в окне *View*. После выхода из подпрограммы симулируйте прерывание еще два раза и пройдите подпрограмму снова (требуется передать три байта информации). В последнем цикле работы подпрограммы устанавливается флаг *F\_trans* и программа переходит в начальное состояние. Таким образом цикл работы программы завершен.

При симуляции не была проверена работа следующих веток программы:

- подавление дребезга контактов при помощи таймера Т0;

- превышение времени таймаута, задаваемого таймером Т1;

- выполнение арифметических операций вычитания, умножения и деления.

Выйдите из симулятора, нажав кнопку Stop Debugging (Ctrl+Shift+F5).

3.5 Запрограммируйте МК стенда и проверьте работу программы на нем. Для этого через, USB-кабель подключите ПК, к COM-порту стенда EasyAVR5. Запустите программу AVRStend. В данной программе можно задавать скорость передачи, а также наличие бит паритета. После каждой смены параметров передачи необходимо нажимать кнопку **Установить**. Далее идет поле ввода данных, после изменения данных необходимо нажать кнопку **Сформировать сообщение**. Новое сообщение появиться в соответствующей строке.

3.6 Выполните передачу команды. Для этого необходимо нажать кнопку **Передать сообщение.** Если стенд подключен правильно, то после передачи должен загореться светодиод СИД. Проверьте правильность принятых данных последовательно нажимая кнопку «Просмотр». Крайний левый индикатор показывает номер просматриваемого байта.

1) Байт кода операции:

- 0x01 – сложение 2-х однобайтных двоичных чисел без знака (0 до 255);

- 0х02 – вычитание 2-х однобайтных двоичных чисел без знака (0 до 255);

- 0х03 – умножение 2-х однобайтных двоичных чисел без знака (0 до 255);

- 0х04 – деление 2-х однобайтных двоичных чисел без знака (0 до 255).

2) Байт первого операнда (1-ое слагаемое, уменьшаемое, множимое, делимое).

3) Байт второго операнда (2-ое слагаемое, вычитаемое, множитель, делитель).

4) Байт контрольной суммы с инверсией

Кадр ответа состоит из следующей последовательности:

5) Старшего байта результата выполнения операции/(частное);

6) Младший байт результата выполнения операции/(остаток от деления);

7) Байт контрольной суммы с инверсией.

3.7 Для передачи ответа на стенде необходимо нажать кнопку «Ответ», при этом в программе AVRStend появится информация в строках **Прием данных** и **Отображение результата**.

3.8 При нажатии кнопки «Просмотр» наблюдается необходимость 1сек паузы между нажатиями. Т.е. время, выбранное для подавления дребезга контактов кнопки выбрано необоснованно большим (рекомендуемое 5мсек). Доработать программу сократив паузу между нажатиями.

3.9 Проведите эксперимент по изменению и передаче команд арифметических операций и правильности их приема стендом EasyAVR5:

- сложение двух чисел, задавая числа в десятичном, двоичном, шестнадцатиричном формате

- вычитание двух чисел, задавая числа в десятичном, двоичном, шестнадцатиричном формате

- умножение двух чисел, задавая числа в десятичном, двоичном, шестнадцатиричном формате

- деление двух чисел, задавая числа в десятичном, двоичном, шестнадцатиричном формате

3.10 Измените скорость передачи в программе МК и программе AVRStend, проверьте и добейтесь правильной работы изменной программы.

3.11 Добавьте бит паритета в программах МК и AVRStend и проверьте и добейтесь правильной работы изменной программы.