МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт компьютерных наук
Кафедра автоматизированных систем управления

Лабораторная работа

по микропроцессорным системам №4

“Последовательные интерфейсы и аналоговые средства микроконтроллеров”

Студент АС-21-1 _____ Станиславчук С. М.

(подпись, дата)

Руководитель

Ст. преподаватель _____ Болдырихин О. В.

(подпись, дата)

Липецк 2024

Содержание

1. Задание, конкретизированное вариантом

Изучить последовательные интерфейсы (TWI, SPI, USART), аналоговый компаратор и аналого-цифровой преобразователь.

Исследователь работу заданного интерфейса. Для этого написать на языке ассемблера программы для двух микроконтроллеров, связанных по данному интерфейсу.

Вариант 7.

Последовательный интерфейс и его режим работы: USART, double speed async, контроль по нечетности.
Режим работы аналого-цифрового преобразователя: Одиночный вход ADC6.
Режим работы аналогового компаратора: Входы — AIN0 и AIN1.

2. Программы

```
; Микроконтроллер №1

.include "m8535def.inc"

.dseg
.equ XTAL = 4000000
.equ baudrate = 9600
.equ bauddivider = XTAL/(16*baudrate)-1
array: .db 0, 0, 0, 0
.equ arr_len = 0x04

.cseg
.org 0x00
rjmp RESET
.org 0x0B
rjmp USART_COMPLETE
.org 0x0E
rjmp ADC_COMPLETE
.org 0x15

RESET:
    ldi r16, LOW(RAMEND)
    out SPL, r16
    ldi r16, HIGH(RAMEND)
    out SPH, r16
    rcall USART_INIT
    rcall ADC_INIT
    ldi r16, 0xFF
    out DDRA, r16
    ldi xl, low(array)
    ldi xh, high(array)
    ldi r17, arr_len
    sei
WAIT:
    rjmp WAIT
```

```
USART_INIT:
    ldi r16, LOW(bauddivider)
    out UBRRH, r16
    ldi r16, HIGH(bauddivider)
    out UBRRL, r16
    ldi r16, 0b10011000
    out UCSRB, r16
    ldi r16, 0
    out UCSRA, r16
    ldi r16, 0b10000110
    out UCSRC, r16
    ret

ADC_INIT:
    ldi r16, 0b11101011
    out ADCSRA, r16
    ldi r16, 0b11100000
    out ADMUX, r16
    ret

ADC_COMPLETE:
    in r16, ADCH
WAIT_UART:
    sbis UCSRA, UDRE
    rjmp WAIT_UART
    out UDR, r16
    reti

USART_COMPLETE:
    in r16, UDR
    st X+, r16
    dec r17
    brne RECEIVED
    rcall SORT
    rcall OUTPUT
    ldi xl, low(array)
```

```
    ldi xh, high(array)
    ldi r17, arr_len
RECEIVED:
    reti

SORT:
    ldi r22, arr_len
MAINLOOP:
    mov r18, r22
    dec r18
    ld xl, r16
    ldi r19, 0
CYCLELOOP:
    ld r20, X+
    ld r21, X
    cp r21, r20
    brsh NOSWAP
    st -X, r21
    st X+, r21
    st X, r20
    ldi r19, 1
NOSWAP:
    dec r18
    brne CYCLELOOP
    cpi r19, 0
    brne MAINLOOP
    ret

OUTPUT:
    ldi r25, arr_len
    ldi r18, 0
    ldi r20, 0
    ldi r21, 0
OUTLOOP:
    ld r19, X+
    out PORTA, r19
    inc r18
```

```
    cpse r18, r25
    brne OUTLOOP
    ret
```

```
; Микроконтроллер №2

.include "m8535def.inc"

.dseg
.equ XTAL = 4000000
.equ baudrate = 9600
.equ bauddivider = XTAL/(16*baudrate)-1
array: .db 0, 0, 0, 0
.equ arr_len = 0x04

.cseg
.org 0x00
rjmp RESET
.org 0x0B
rjmp USART_COMPLETE
.org 0x10
rjmp CMP_COMPLETE
```

```
.org 0x15

RESET:
    ldi r16, LOW(RAMEND)
    out SPL, r16
    ldi r16, HIGH(RAMEND)
    out SPH, r16
    rcall USART_INIT
    ldi r16, 0xFF
    out DDRA, r16
    ldi r16, 0b01001011
    out ACSR, r16
    ldi xl, low(array)
    ldi xh, high(array)
    ldi r17, arr_len
    ldi r23, 0
    ldi r24, arr_len
    sei
WAIT:
    rjmp WAIT

USART_INIT:
    ldi r16, LOW(bauddivider)
    out UBRRL, r16
    ldi r16, HIGH(bauddivider)
    out UBRRH, r16
    ldi r16, 0b10011000
    out UCSRB, r16
    ldi r16, 0
    out UCSRA, r16
    ldi r16, 0b10000110
    out UCSRC, r16
    ret

CMP_COMPLETE:
    in r17, ACSR
    lsl r17
```

```
    or r23, r17
    dec r24
    brne CONT_WAIT
WAIT_UART:
    sbis UCSRA, UDRE
    rjmp WAIT_UART
    out UDR, r16
CONT_WAIT:
    reti

USART_COMPLETE:
    in r16, UDR
    st X+, r16
    dec r17
    brne RECEIVED
    rcall SORT
    rcall OUTPUT
RECEIVED:
    reti

SORT:
    ldi r22, arr_len
MAINLOOP:
    mov r18, r22
    dec r18
    ldi xl, low(array)
    ldi xh, high(array)
    ldi r19, 0
CYCLELOOP:
    ld r20, X+
    ld r21, X
    cp r21, r20
    brsh NOSWAP
    st -X, r21
    st X+, r21
    st X, r20
    ldi r19, 1
```

```
NOSWAP:
    dec r18
    brne CYCLELOOP
    cpi r19, 0
    brne OUTLOOP

OUTPUT:
    ldi r25, arr_len
    ldi r18, 0
    ldi r20, 0
    ldi r21, 0
OUTLOOP:
    ld r19, X+
    out PORTA, r19
    inc r18
    cpse r18, r25
    brne OUTLOOP
    ret
```

Листинг программ

```
; Микроконтроллер №1

.dseg
.equ XTAL = 4000000
.equ baudrate = 9600
.equ bauddivider = XTAL/(16*baudrate)-1
000060 00
000061 00
000062 00
000063 00 array: .db 0, 0, 0, 0
.equ arr_len = 0x04

.cseg
.org 0x00
000000 c014 rjmp RESET
.org 0x0B
00000b c02b rjmp USART_COMPLETE
.org 0x0E
00000e c023 rjmp ADC_COMPLETE
.org 0x15
RESET:
000015 e50f ldi r16, LOW(RAMEND)
000016 bf0d out SPL, r16
000017 e002 ldi r16, HIGH(RAMEND)
000018 bf0e out SPH, r16
000019 d008 rcall USART_INIT
00001a d012 rcall ADC_INIT
00001b ef0f ldi r16, 0xFFout DDRA, r16
00001d e6a0 ldi xl, low(array)
00001e e0b0 ldi xh, high(array)
00001f e014 ldi r17, arr_len
000020 9478 sei
WAIT:
000021 cfff rjmp WAIT
USART_INIT:
```

```
000022 e109 ldi r16, LOW(bauddivider)
000023 b909 out UBRRL, r16
000024 e000 ldi r16, HIGH(bauddivider)
000025 bd00 out UBRRH, r16
000026 e908 ldi r16, 0b10011000
000027 b90a out UCSRB, r16
000028 e000 ldi r16, 0
000029 b90b out UCSRA, r16
00002a e806 ldi r16, 0b10000110
00002b bd00 out UCSRC, r16
00002c 9508 ret
ADC_INIT:
00002d ee0b ldi r16, 0b11101011
00002e b906 out ADCSRA, r16
00002f ee00 ldi r16, 0b11100000
000030 b907 out ADMUX, r16
000031 9508 ret
ADC_COMPLETE:
000032 b105 in r16, ADCH
WAIT_UART:
000033 9b5d sbis UCSRA, UDRE
000034 cffe rjmp WAIT_UART
000035 b90c out UDR, r16
000036 9518 reti
USART_COMPLETE:
000037 b10c in r16, UDR
000038 930d st X+, r16
000039 951a dec r17brne RECEIVED
00003b d005 rcall SORT
00003c d017 rcall OUTPUT
00003d e6a0 ldi xl, low(array)
00003e e0b0 ldi xh, high(array)
00003f e014 ldi r17, arr_len
RECEIVED:
000040 9518 reti
SORT:
000041 e064 ldi r22, arr_len
```

```
000042 2f26 MAINLOOP: mov r18, r22
000043 952a dec r18
000044 e6a0 ldi xl, low(array)
000045 e0b0 ldi xh, high(array)
000046 e030 ldi r19, 0
000047 914d CYCLELOOP: ld r20, X+
000048 915c ld r21, X
000049 1754 cp r21, r20
00004a f420 brsh NOSWAP
00004b 935e st -X, r21
00004c 935d st X+, r21
00004d 934c st X, r20
00004e e031 ldi r19, 1
00004f 952a NOSWAP: dec r18
000050 f7b1 brne CYCLELOOP
000051 3030 cpi r19, 0
000052 f779 brne MAINLOOP
000053 9508 ret
OUTPUT:
000054 e094 ldi r25, arr_len
000055 e020 ldi r18, 0
000056 e040 ldi r20, 0
000057 e050 ldi r21, 0
OUTLOOP:
000058 913d ld r19, X+
000059 bb3b out PORTA, r19
00005a 9523 inc r18
00005b 1329 cpse r18, r25
00005c f7d9 brne OUTLOOP
00005d 9508 ret
```

```
; Микроконтроллер №2

000063 00 array: .db 0, 0, 0, 0
.equ arr_len = 0x04

.cseg
.org 0x00
000000 c014 rjmp RESET
.org 0x0B
00000b c02e rjmp USART_COMPLETE
.org 0x10
000010 c01f rjmp CMP_COMPLETE
.org 0x15

RESET:
000015 e50f ldi r16, LOW(RAMEND)
000016 bf0d out SPL, r16
000017 e002 ldi r16, HIGH(RAMEND)
000018 bf0e out SPH, r16
000019 d00b rcall USART_INIT
00001a ef0f ldi r16, 0xFF
00001b bb0a out DDRA, r16
00001c e40b ldi r16, 0b01001011
00001d b908 out ACSR, r16
00001e e6a0 ldi xl, low(array)
00001f e0b0 ldi xh, high(array)
000020 e014 ldi r17, arr_len
000021 e070 ldi r23, 0
000022 e084 ldi r24, arr_len
000023 9478 sei
WAIT:
000024 cfff rjmp WAIT

USART_INIT:
```

```
000025 e109 ldi r16, LOW(bauddivider)
000026 e000 ldi r16, HIGH(bauddivider)
000027 bd00 out UBRRH, r16
000028 e908 ldi r16, 0b10011000
000029 b90a out UCSRB, r16
00002a e000 ldi r16, 0
00002b b90b out UCSRA, r16
00002c e806 ldi r16, 0b10000110
00002d bd00 out UCSRC, r16
00002e 9508 ret

CMP_COMPLETE:
000030 b110 in r17, ACSR >> 5
000031 7011 andi r17, 1
000032 0f77 lsl r23
000033 2b71 or r23, r17
000034 958a dec r24
000035 f419 brne CONT_WAIT
WAIT_UART:
000036 9b5d sbis UCSRA, UDRE
000037 cffe rjmp WAIT_UART
000038 b90c out UDR, r16
CONT_WAIT:
000039 9518 reti

USART_COMPLETE:
00003a b10c in r16, UDR
00003b 930d st X+, r16
00003c 951a dec r17
00003d f411 brne RECEIVED
00003e d002 rcall SORT
00003f d014 rcall OUTPUT
RECEIVED:
000040 9518 reti

SORT:
000041 e064 ldi r22, arr_len
```

```
000042 2f26 MAINLOOP: mov r18, r22
000043 952a dec r18
000044 e6a0 ldi xl, low(array)
000045 e0b0 ldi xh, high(array)
000046 914d CYCLELOOP: ld r20, X+
000047 915c ld r21, X
000048 1754 cp r21, r20
000049 f420 brsh NOSWAP
00004a 935e st -X, r21
00004b 935d st X+, r21
00004c 934c st X, r20
00004d e031 ldi r19, 1
00004e 952a NOSWAP: dec r18
00004f f7b1 brne CYCLELOOP
000050 3030 cpi r19, 0
000051 f779 brne MAINLOOP
000052 9508 ret

OUTPUT:
000054 e094 ldi r25, arr_len
000055 e020 ldi r18, 0
000056 e040 ldi r21, 0

OUTLOOP:
000058 913d ld r19, X+
000059 bb3b out PORTA, r19
00005a 9523 inc r18
00005b 1329 cpse r18, r25
00005c f7d9 brne OUTLOOP
00005d 9508 ret
```

3. Вывод

В данной лабораторной работе рассмотрен процесс взаимодействия между двумя микроконтроллерами по последовательному интерфейсу. Микроконтроллер ATmega8535 имеет в своем составе модуль универсального синхронно/асинхронного приемопередатчика - USART. С его помощью между компьютером и микроконтроллером (либо между двумя микроконтроллерами) можно организовать обмен данными по последовательному каналу. Инициализация порта USART выполняется с помощью установки битов в портах UBRRH:UBRRL, UCSRA, UCSRB, UCSRC. В первых двух из них задается значение скорости передачи данных. Принятый символ USART модуль сохраняет в регистре данных UDR. Оттуда его можно переписывать в буфер. Выполняется это действие в прерывании, для этого оно должно быть разрешено.

Также в составе микроконтроллера AVR присутствует аналоговый компаратор. Он сравнивает между собой два напряжения и заносит результат сравнения в регистр. Также он может вызывать прерывания, если результат сравнения изменился, и управлять схемой захвата таймера T1.

Аналогово-цифровой преобразователь преобразует некоторый аналоговый сигнал в цифровой. На вход АЦП подается непрерывный аналоговый сигнал, а на выходе получается последовательность цифровых значений. Принцип работы АЦП основывается на сравнении полученного уровня напряжения с опорным, которое всегда должно иметь большее значение, чем измеряемое. В микроконтроллере AVR ATmega8535 АЦП имеет разрешение 10 бит.