**REQUIREMENTS NOT MET**

N/A

**PROBLEMS ENCOUNTERED**

N/A

**FUTURE WORK/APPLICATIONS**

<insert a brief paragraph describing how the topics covered in this lab could potentially be used for other applications>

THE ABOVE SHOULD BE LIMITED TO THE FIRST PAGE, AND NOTHING ELSE SHOULD BE INCLUDED, WHICH ALSO IMPLIES THAT THIS SENTENCE OF TEXT SHOULD BE REMOVED.

**PRE-LAB EXERCISES**

**i. The sampling rate of a UART receiver is usually faster than the baud rate of the overall system. Why is this so?   
 The receivers needs to be able to read data at any time since its asynchronous. In addition, 3 samples need to be taken in order for the clock synchronizer to determine if it’s a legitimate start bit or just noise. This requires a faster sampling rate on the receiver.**

**ii. What is the maximum possible baud rate for asynchronous communication within the USART system of the ATxmega128A1U, assuming that the microcontroller has a system clock frequency of 2 MHz and that the USART “double-speed mode” is disabled (i.e., the relevant bit CLK2X is set to 0)? In addition to the maximum rate, provide the values of the relevant registers used to configure that rate. Whenever appropriate, support your answer with calculations.   
 BAUDCTRLA would need to be 0b00000000  
 BAUDCTRLB would need to be 0b00000000** A picture containing text, font, handwriting, white

Description automatically generated **Figure 1: Baud rate calculations**

**iii. In the context of the USART system within the ATxmega128A1U, how many buffers (i.e., memory locations that store temporary data) are used by a transmitter? How many are used by a receiver? Additionally, for both transmitters and receivers, explain how the use of buffers provides greater flexibility to an application involving these components.**

**The transmitter contains 2 registers:  
1: The shift register, which is used for transmitting receiving data  
2. The data register, which is used for holding data while the shift register receives or transmits data**

**The receiver contains 3 registers  
1: The shift register, which is used for receiving data  
2 and 3: The data register, which is used for holding data for parity checking and synchronization while the shift register receives data  
  
The buffers allow data to be received/transmitted while being written to/read from by the microcontroller.**

**iv. If an asynchronous serial communication protocol of 7 data bits, one start bit, one stop bits, odd parity, and baud rate of 15.6 kHz was chosen, calculate how many seconds it would take to transmit the ASCII character string “Dr. Schwartz saw seven slick slimy snakes slowly sliding southward.” (This string has 67 characters.) Note that ASCII is a 7-bit (not an 8-bit) code. Show all work.**

A picture containing text, font, handwriting, screenshot

Description automatically generated

**Figure 2: Baud rate calculations**

**PSEUDOCODE/FLOWCHARTS**

**SECTION 2**

**A picture containing text, diagram, plan, parallel

Description automatically generated  
Figure 3: Flowchart for “lab5\_2.asm”**

**SECTION 3**

**SECTION 4**

**A picture containing text, diagram, plan, technical drawing

Description automatically generated**

**Figure 5: Flowchart for “lab5\_4.asm”**

**SECTION 5**

**A picture containing text, diagram, plan, parallel

Description automatically generated**

**Figure 6: Flowchart for “lab5\_5.asm”**

**SECTION 6**

**A picture containing text, diagram, plan, technical drawing

Description automatically generated**

**Figure 7: Flowchart for “lab5\_6.asm”**

**SECTION 7**

**PROGRAM CODE**

**SECTION 2**

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;Lab 5, Section 2

;Name: Steven Miller

;Class #: 11318

;PI Name: Anthony Stross

;Description: transmits character over serial usb line

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*INCLUDES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.include "ATxmega128a1udef.inc"

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END OF INCLUDES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*EQUATES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.equ bsel = 47

.equ bscale = -6

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END OF EQUATES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DEFS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END OF DEFS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*PROGRAM MEMORY CONFIGURATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*END OF PROGRAM MEMORY CONFIGURATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*DATA MEMORY CONFIGURATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*END OF DATA MEMORY CONFIGURATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*MAIN PROGRAM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.CSEG

.org 0x0000

rjmp main

.org 0x0200

main:

;initialize stack

ldi r16, low(0x3fff)

out CPU\_SPL, r16

ldi r16, high(0x3fff)

out CPU\_SPH, r16

rcall USART\_INIT

loop:

ldi r16,'U'

rcall OUT\_CHAR

rjmp loop

end:

rjmp end

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Name: USART\_INIT

; Purpose: INITIALIZE USART MODULE ON PORT D

; Input(s): N/A

; Output: N/A

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

USART\_INIT:

push r16

;set port d pin 3 as output

ldi r16, 0b00001000

sts PORTD\_OUTSET,r16

sts PORTD\_DIRSET, r16

;enable transmitter

ldi r16, 0b00001000

sts USARTD0\_CTRLB, r16

;set transmission to asynchronous and parity to odd

;and set number of stop bits to 1 and set character size to 8 bits

ldi r16,(USART\_PMODE\_ODD\_gc|USART\_CMODE\_ASYNCHRONOUS\_gc|USART\_CHSIZE\_8BIT\_gc)

sts USARTD0\_CTRLC, r16

;set baud rate to 72000 bps

;bsel = 47

;bscale = -6

ldi r16, low(bsel)

sts USARTD0\_BAUDCTRLA, r16

ldi r16, ((bscale<<4) | (high(bsel))) ;1010 = -6

sts USARTD0\_BAUDCTRLB, r16

pop r16

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Name: OUT\_CHAR

; Purpose:TRANSMIT CHARACTER OUT OF PORT D TO USB

; Input(s): N/A

; Output: USARTD0\_DATA

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

OUT\_CHAR:

push r17

;check if transmitter busy

transmitter\_busy:

lds r17, USARTD0\_STATUS

sbrs r17, USART\_DREIF\_bp

rjmp transmitter\_busy

;ldi r17, 0b0

;load character into transmitter data register

sts USARTD0\_DATA, r16

pop r17

ret

**SECTION 3**

**SECTION 4**

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;Lab 5, Section 4

;Name: Steven Miller

;Class #: 11318

;PI Name: Anthony Stross

;Description: transmits character string over serial usb line

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*INCLUDES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.include "ATxmega128a1udef.inc"

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END OF INCLUDES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*EQUATES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.equ bsel = 47

.equ bscale = -6

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END OF EQUATES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DEFS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END OF DEFS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*PROGRAM MEMORY CONFIGURATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.CSEG

.ORG 0X3744

TABLE:

.DB "STEVEN MILLER"

;\*\*\*\*\*\*\*\*\*\*\*END OF PROGRAM MEMORY CONFIGURATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*DATA MEMORY CONFIGURATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*END OF DATA MEMORY CONFIGURATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*MAIN PROGRAM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.CSEG

.org 0x0000

rjmp main

.org 0x0200

main:

;initialize stack

ldi r16, low(0x3fff)

out CPU\_SPL, r16

ldi r16, high(0x3fff)

out CPU\_SPH, r16

rcall USART\_INIT

;set z pointer

ldi ZL, byte1(TABLE<<1)

ldi ZH, byte2(table<<1)

ldi r16, byte3(table<<1)

sts CPU\_RAMPZ, r16

rcall OUT\_STRING

end:

rjmp end

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Name: USART\_INIT

; Purpose: INITIALIZE USART MODULE ON PORT D

; Input(s): N/A

; Output: N/A

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

USART\_INIT:

push r16

;set port d pin 3 as output

ldi r16, 0b00001000

sts PORTD\_OUTSET,r16

sts PORTD\_DIRSET, r16

;enable transmitter

ldi r16, 0b00001000

sts USARTD0\_CTRLB, r16

;set transmission to asynchronous and parity to odd

;and set number of stop bits to 1 and set character size to 8 bits

ldi r16,(USART\_PMODE\_ODD\_gc|USART\_CMODE\_ASYNCHRONOUS\_gc|USART\_CHSIZE\_8BIT\_gc)

sts USARTD0\_CTRLC, r16

;set baud rate to 72000 bps

;bsel = 47

;bscale = -6

ldi r16, low(bsel)

sts USARTD0\_BAUDCTRLA, r16

ldi r16, ((bscale<<4) | (high(bsel))) ;1010 = -6

sts USARTD0\_BAUDCTRLB, r16

pop r16

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Name: OUT\_CHAR

; Purpose:TRANSMIT CHARACTER OUT OF PORT D TO USB

; Input(s): N/A

; Output: USARTD0\_DATA

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

OUT\_CHAR:

push r17

;check if transmitter busy

transmitter\_busy:

lds r17, USARTD0\_STATUS

sbrs r17, USART\_DREIF\_bp

rjmp transmitter\_busy

;ldi r17, 0b0

;load character into transmitter data register

sts USARTD0\_DATA, r16

pop r17

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Name: OUT\_STRING

; Purpose:TRANSMIT STRING OUT OF PORT D TO USB

; Input(s): N/A

; Output:N/A

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

OUT\_STRING:

push r16

read\_string:

;read character pointed to by z

elpm r16, z+

;check if null

cpi r16, 0x00

breq null

not\_null:

rcall OUT\_CHAR

rjmp read\_string

null:

pop r16

ret

**SECTION 5**

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;Lab 5, Section 5

;Name: Steven Miller

;Class #: 11318

;PI Name: Anthony Stross

;Description: recieves serial data from computer

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*INCLUDES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.include "ATxmega128a1udef.inc"

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END OF INCLUDES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*EQUATES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.equ bsel = 47

.equ bscale = -6

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END OF EQUATES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DEFS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END OF DEFS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*PROGRAM MEMORY CONFIGURATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*END OF PROGRAM MEMORY CONFIGURATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*DATA MEMORY CONFIGURATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*END OF DATA MEMORY CONFIGURATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*MAIN PROGRAM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.CSEG

.org 0x0000

rjmp main

.org 0x0200

main:

;initialize stack

ldi r16, low(0x3fff)

out CPU\_SPL, r16

ldi r16, high(0x3fff)

out CPU\_SPH, r16

rcall USART\_INIT

loop:

rcall IN\_CHAR

rcall OUT\_CHAR

rjmp loop

end:

rjmp end

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Name: USART\_INIT

; Purpose: INITIALIZE USART MODULE ON PORT D

; Input(s): N/A

; Output: N/A

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

USART\_INIT:

push r16

;set port d pin 2 as input and port d pin 3 as output

ldi r16, 0b00000100

sts PORTD\_OUTCLR,r16

sts PORTD\_DIRCLR, r16

ldi r16, 0b00001000

sts PORTD\_OUTSET,r16

sts PORTD\_DIRSET, r16

;enable transmitter and reciever

ldi r16, 0b00011000

sts USARTD0\_CTRLB, r16

;set transmission to asynchronous and parity to odd

;and set number of stop bits to 1 and set character size to 8 bits

ldi r16,(USART\_PMODE\_ODD\_gc|USART\_CMODE\_ASYNCHRONOUS\_gc|USART\_CHSIZE\_8BIT\_gc)

sts USARTD0\_CTRLC, r16

;set baud rate to 72000 bps

;bsel = 47

;bscale = -6

ldi r16, low(bsel)

sts USARTD0\_BAUDCTRLA, r16

ldi r16, ((bscale<<4) | (high(bsel))) ;1010 = -6

sts USARTD0\_BAUDCTRLB, r16

pop r16

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Name: OUT\_CHAR

; Purpose:TRANSMIT CHARACTER OUT OF PORT D TO USB

; Input(s): N/A

; Output: USARTD0\_DATA

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

OUT\_CHAR:

push r17

;check if transmitter busy

transmitter\_busy:

lds r17, USARTD0\_STATUS

sbrs r17, USART\_DREIF\_bp

rjmp transmitter\_busy

;load character into transmitter data register

sts USARTD0\_DATA, r16

pop r17

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Name: IN\_CHAR

; Purpose: RECIEVES CHARACTER FROM USB TO RECIEVER

; Input(s):USARTD0\_DATA

; Output:N/A

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IN\_CHAR:

push r17

;check if reciever busy

reciever\_busy:

lds r17, USARTD0\_STATUS

sbrs r17, USART\_RXCIF\_bp

rjmp reciever\_busy

;load character into reciever data register

lds r16, USARTD0\_DATA

pop r17

ret

**SECTION 6**

**SECTION 7**

**APPENDIX**

<insert copy of all *supporting* ASM or C program code, e.g., header files referenced within your programs, as well as any other relevant information, e.g., screenshots (with meaningful captions), when applicable (if not applicable, write “N/A”)>