Patrick Austin

CPE 301 - 1104, Fall 2016

Homework 6

10/17/2016

1. -use normal transmission speed

Set UCSR0A bit 1 to 0.

-disable multiprocessor communication mode

Set UCSR0A bit 0 to 0.

-turn off all USART0 interrupts

Set UCSR0B bits 5, 6, 7, to 0

-turn off the receiver and turn on the transmitter

Set UCSR0B bits 3, 4 to 1

-set the character size to 8 bits

Set UCSR0C bits 1, 2 to 1

-use asynchronous USART mode

Set UCSR0C bits 6, 7 to 0

-use no parity

Set UCSR0C bits 4, 5 to 0

-use 1 stop bit

Set UCSR0C bit 3 to 0

-set baud rate to 115200 bits per seconds

Place value 0d8 in UBRRH and UBRRL registers.

2 & 3. The functions are implemented in the attached program.

//Patrick Austin

//CPE 301 Homework 6, problems 2 & 3

//Revision Number 1

//Revision date: 10/17/2016

//global hardware pointers

volatile unsigned char\* myUCSR0A = (unsigned char\*) 0xC0;

volatile unsigned char\* myUCSR0B = (unsigned char\*) 0xC1;

volatile unsigned char\* myUCSR0C = (unsigned char\*) 0xC2;

volatile unsigned int\* myUBBR0 = (unsigned int\*) 0xC4;

volatile unsigned char\* myUDR0 = (unsigned char\*) 0xC6;

//function prototypes

//initializes the UART as specified in problem 1

void U0init();

//returns non-zero if RDA status bit is set

unsigned char U0kbhit();

//returns char from UART input buffer

unsigned char U0getchar();

//loads char to UART transmit buffer

void U0putchar(unsigned char U0pdata);

void setup()

{

//stub: only implementing functions for this homework

}

void loop()

{

//stub: only implementing functions for this homework

}

//function implementations

void U0init()

{

\*myUCSR0A = 0x20; //set UDRE high, all others low

\*myUCSR0B = 0x18; //set receiver enable and transmitter enable high, others low

\*myUCSR0C = 0x06; //set 8 bit char size, asynchronous, no parity, 1 bit stop

\*myUBBR0 = (16000000 / 16 / 115200 - 1); //calculate UBRR value for 115200 baud.

//calculates to 0d8 for 115200 baud

//which matches the datasheet

}

unsigned char U0kbhit()

{

return (\*myUCSR0A & 0x80); //returns 0x80 if RDA status bit is set and 0 if not

}

unsigned char U0getchar()

{

return (\*myUDR0); //return the character in the input buffer

}

void U0putchar(unsigned char U0pdata)

{

while ( (\*myUCSR0A & 0x20) == 0 ) //until USART0 TBE set, do nothing

{}

\*myUDR0 = U0pdata; //once set, put char in the transmit buffer

