Patrick Austin

CPE 301 – 1104

Assignment # 8

11/6/2016

Assignment description:

In this lab we implemented several basic serial functions for use in a program that would send a char from the PC keyboard to the Arduino and then mirror that char back to the PC’s serial terminal. Then we created an elaboration on that program that read a char from the keyboard as a serial input, decoded its bits, and sent back the ASCII value for the character in hexadecimal.

Problems encountered:

Much of this lab was a review of in-class and homework exercises on using serial with the Arduino, so there were not many difficulties. It took a little discussion within the group to decide on an elegant way to decode the byte into nibbles and to equate those nibbles to corresponding ASCII values. My biggest difficulties came in dealing with Putty, which is temperamental on my machine and often requires a restart to work correctly. No diagnosis for this bug has been forthcoming, which has been a minor annoyance but did not impede my ultimate ability to complete the lab correctly.

Lessons learned:

Since we are fairly far ahead of this lab in lecture, the material was largely a review on serial, the basic serial functions, and manipulating those functions to do basic tasks (the mirroring in the first program) or slightly more complicated ones (the decoding in the second program). Got more hands-on experience with these topics as well as using Putty to interface with the board, which was welcome.

Description of completed lab:

Here is the code for Echo2:

//Patrick Austin

//CPE 301 Lab 8: echo2c

//Revision Number 1

//Revision date: 11/7/2016

//NOTE: adapted from class website code "echo3C.txt" as specified in the lab prompt

//as well as homework

volatile unsigned char \*myUCSR0A = (unsigned char \*)0x00C0;

volatile unsigned char \*myUCSR0B = (unsigned char \*)0x00C1;

volatile unsigned char \*myUCSR0C = (unsigned char \*)0x00C2;

volatile unsigned int \*myUBRR0 = (unsigned int \*) 0x00C4;

volatile unsigned char \*myUDR0 = (unsigned char \*)0x00C6;

void setup()

{

U0init(9600);

}

void loop()

{

unsigned char cs1;

while (U0kbhit()==0){}; // wait for RDA = true

cs1 = U0getchar(); // read character

U0putchar(cs1); // echo character

}

void U0init(int U0baud)

{

unsigned long FCPU = 16000000;

unsigned int tbaud;

tbaud = (FCPU / 16 / U0baud - 1);

\*myUCSR0A = 0x20;

\*myUCSR0B = 0x18;

\*myUCSR0C = 0x06;

\*myUBRR0 = tbaud;

}

unsigned char U0kbhit()

{

return (\*myUCSR0A & 0x80);

}

unsigned char U0getchar()

{

return (\*myUDR0);

}

void U0putchar(unsigned char U0pdata)

{

while ( (\*myUCSR0A & 0x20) == 0 )

{}

\*myUDR0 = U0pdata;

}

Here is a screenshot of Echo2 compiling successfully:



Here is a screenshot of Echo2’s output in Putty with input ‘abcde12345’:



Here is the code for Echo3:

//Patrick Austin

//CPE 301 Lab 8: echo3c

//Revision Number 1

//Revision date: 11/7/2016

//NOTE: adapted from class website code "echo3C.txt" as specified in the lab prompt

//as well as homework

volatile unsigned char \*myUCSR0A = (unsigned char \*)0x00C0;

volatile unsigned char \*myUCSR0B = (unsigned char \*)0x00C1;

volatile unsigned char \*myUCSR0C = (unsigned char \*)0x00C2;

volatile unsigned int \*myUBRR0 = (unsigned int \*) 0x00C4;

volatile unsigned char \*myUDR0 = (unsigned char \*)0x00C6;

//lookup table containing corresponding ASCII values to index, such that asciiValues[i]

//is the ASCII code for i in hexadecimal, for a range from 0-F

static const unsigned char asciiValues[] =

{ 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 65, 66, 67, 68, 69, 70};

void setup()

{

U0init(9600);

}

void loop()

{

unsigned char cs1;

unsigned char output1;

unsigned char output2;

while (U0kbhit()==0){}; // wait for RDA = true

cs1 = U0getchar(); // read character

output1 = cs1 >> 4; //get top 4 bits

output2 = cs1 & 0x0F; //get bottom 4 bits

U0putchar('0'); // send 0

U0putchar('x'); // send x

U0putchar( char(asciiValues[output1]) ); //convert top 4 bits to ASCII

//equivalents and send

U0putchar( char(asciiValues[output2]) ); //convert bottom 4 bits to ASCII eq and send

U0putchar('\n'); //send an endline char

}

void U0init(int U0baud)

{

unsigned long FCPU = 16000000;

unsigned int tbaud;

tbaud = (FCPU / 16 / U0baud - 1);

\*myUCSR0A = 0x20;

\*myUCSR0B = 0x18;

\*myUCSR0C = 0x06;

\*myUBRR0 = tbaud;

}

unsigned char U0kbhit()

{

return (\*myUCSR0A & 0x80);

}

unsigned char U0getchar()

{

return (\*myUDR0);

}

void U0putchar(unsigned char U0pdata)

{

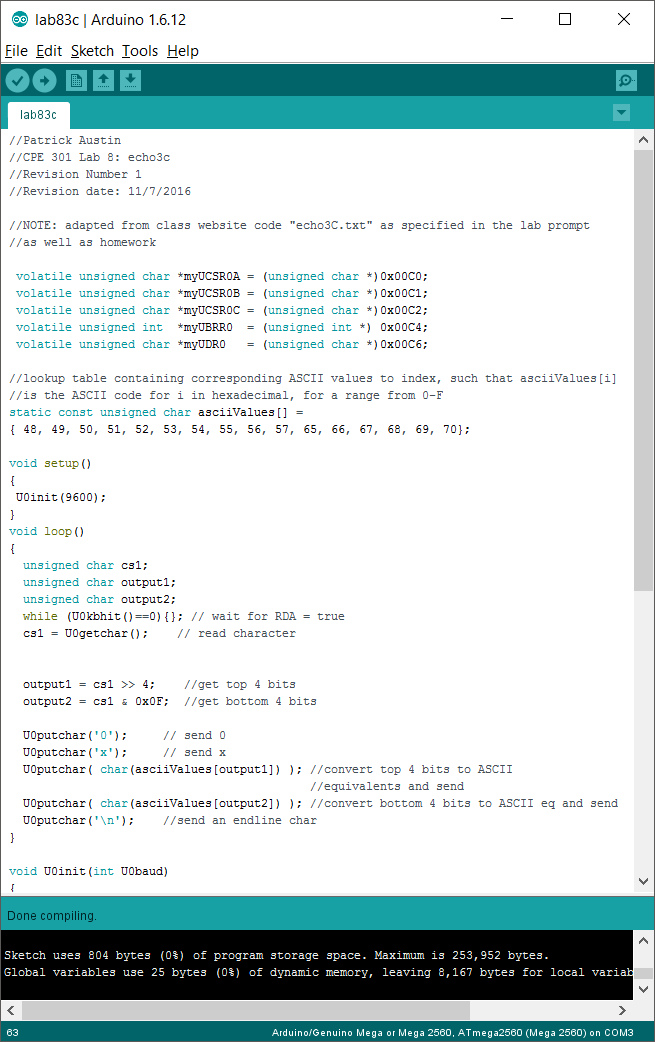
while ( (\*myUCSR0A & 0x20) == 0 )

{}

\*myUDR0 = U0pdata;

}

Here is a screenshot of Echo3 compiling successfully:



Here is a screenshot of Echo2’s output in Putty with input ‘abcdef12345’:

