

Homework Assignment #3

EE475

Fall 2003

Assigned: Thursday, October 2, 2003

Due AT THE START OF CLASS on Thursday, October 16, 2003

You need to hand in a concisely written description of your approach, your actual source code including comments, a summary of program output, and any work you did to check and verify the results.

An embedded system has created a *binary* data file. The file first contains a two byte integer (stored in binary), followed by a sequence of binary data values.

The initial most significant byte is:

1	1	1	1	1	1	1	0
---	---	---	---	---	---	---	---

(0xFE)

and the least significant byte is:

0	1	0	1	0	1	<i>b1</i>	<i>b0</i>
---	---	---	---	---	---	-----------	-----------

(0x54+b1:b0)

The least significant two bits (b0 b1) indicate how to interpret the *rest* of the file:

<i>b1</i>	<i>b0</i>	
0	0	2 byte (short) int
0	1	4 byte (long) int
1	0	4 byte float
1	1	8 byte double

The initial two bytes are stored in the file as a two byte short integer, so if the data is big endian the 0xFE is the first byte, while if the data is little endian the 0xFE is the second byte.

What you need to do:

- Write a C program that can open and read a binary data file as specified above. The program must determine whether the data file is stored little or big endian, and identify whether the rest of the file contains shorts, longs, floats, or doubles.
- The program must determine if the endian order of the file matches the endian order of the processor running the program.
- The program must read each binary value from the file, re-order the bytes to match the endian flavor of the processor, and print the data to the screen or to an output text file. The data should show a recognizable pattern if you converted the file correctly.

You must submit an explanation of your approach, your *commented C* code, your verification procedure, and a summary of the results for each of the 4 test files from the class web site.