Sabreena Abedin (sa7cx) 2/16/2016 floatingpoint.pdf

Your magic (32 bit) floating point number is -2.38671875

This is the number that needs to be converted to (little endian) binary, and expressed in hexadecimal.

Sign bit: 1 (negative)

2.38671875 / 2^1 = 1.193359375

Exponent: 1 + 127 = 128

➔ binary: 1000 0000

Mantissa: 1.193359375

.193359375 – (1/8) = .068359375

.068359375 – (1/16) = .005859375

.005859375 – (1/256) = .001953125

.001953125 – (1/512) = 0

0011 0001 1000 0000 0000 000

Binary Answer: 1100 0000 0001 1000 1100 0000 0000 0000

12 0 1 8 12 0 0 0

Hexadecimal Answer: 0xc018c000

Little Endian Binary Answer: 0000 0000 0000 0011 0010 0100 0000 0011

Little Endian Hexadecimal Answer: 0x00c018c0

Your other magic floating point number is, in hex, 0x0040203f

This is the number that needs to be converted to a (32 bit) floating point number.

Note that the hexadecimal printed above is in little-endian format!

1. Big-endian: 0x3f204000
2. Binary:
   3: 0011
   f: 1111
   2: 0010
   0: 0000
   4: 0100
   0: 0000
   0: 0000
   0: 0000
   = 0011 1111 0010 0000 0100 0000 0000 0000
3. Sign bit: 0 (positive)
   Exponent: 011 1111 0
   = 1 * 2^6 + 1 * 2 ^5 + 1 * 2^ 4 + 1 * 2 ^3 + 1 * 2^2 + 1 * 2 ^1
   = 126
   ➔ 126 – 127 = -1
   Mantissa: 010 0000 0100 0000 0000 0000
   = (1/4) + (1/512) = .251953125
   ➔ 1.251953125

   1.251953125 * 2^-1 = .6259765625

Floating Point Answer: 0.6259765625