Week 1 – Bits & Bytes

Student number: 569091

Contents

Contents 1
Assignment 1.1: Bits & Bytes intro 2
What are Bits & Bytes? 2
What is a nibble? 2
What relationship does a nibble have with a hexadecimal value? 2
Why is it wise to display binary data as hexadecimal values? 2
What kind of relationship does a byte have with a hexadecimal value? 2
An IPv4 subnet is 32-bit, show with a calculation why this is the case. 3
Assignment 1.2: Your favourite color 4
Assignment 1.3: Manipulating binary data 4
Assignment 1.4: Student number to HEX and Binary 5

Assignment 1.1: Bits & Bytes intro

What are Bits & Bytes?

Bit: A bit (binary digit) is the smallest unit of data in computing and can have a value of either 0 or 1 (e.g. false/true or off/on).

Byte: A byte is a collection of 8 bits. It's the standard unit used to represent a single character in memory (such as a letter, number, or symbol) in most computing systems.

What is a nibble?

Nibble: A group of 4 bits (half a byte).

What relationship does a nibble have with a hexadecimal value?

A nibble is a group of 4 bits, capable of representing 2^4 = 16 distinct values, ranging from 0 to 15 in decimal. Each hexadecimal digit (0-F) also represents exactly 16 values.

Why is it wise to display binary data as hexadecimal values?

Readability: Binary numbers are long and difficult for humans to read. Hexadecimal numbers shorten the representation.

Compactness: Hexadecimal reduces the length of binary strings by grouping every 4 binary bits into one hex digit, making complex binary sequences easier to manage

What kind of relationship does a byte have with a hexadecimal value?

In hexadecimal, a byte is represented as two hex digits. Each hexadecimal digit covers a nibble (4 bits), so two hex digits combine to represent a byte (8 bits). For example, the binary byte 1111 1111 is represented as FF in hexadecimal.

Summary:

A byte contains 8 bits and is written as 2 hexadecimal digits, because each hexadecimal digit represents 4 bits (one nibble).

An IPv4 subnet is 32-bit, show with a calculation why this is the case.

An IPv4 address is 32 bits because it has 4 octets, each 8 bits. 4 x 8 = 32. The subnet mask also 32 bits following the same structure.

Example (255.0.0.0), written as 4 bytes (4 × 8 bits = 32 bits):

255 in binary is 11111111, 0 is 00000000

So, the mask is: 11111111.00000000.00000000.00000000. Which totals to 32 bits.

255 decimal to binary		
Bit number	**calculation**	**value**
8	1*2^7	128
7	1*2^6	64
6	1*2^5	32
5	1*2^4	16
4	1*2^3	8
3	1*2^2	4
2	1*2^1	2
1	1*2^0	1
8 total	11111111 total	255 total

Assignment 1.2: Your favourite color

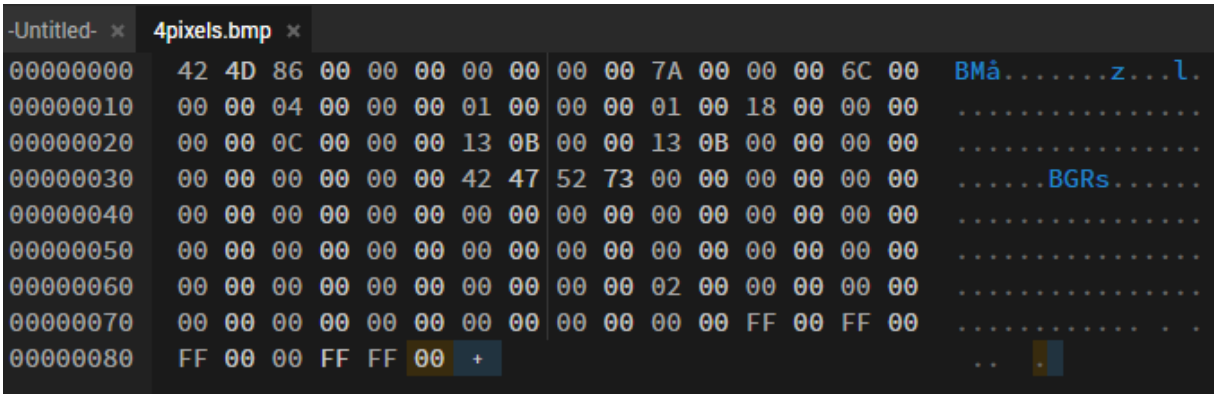
Hexadecimal color code: #00ffff

Screenshot from paint with the image resized to 40x10 and fully zoomed in:

Assignment 1.3: Manipulating binary data

Colour	Colour code hexadecimaal (RGB)	Big Endian	Little Endian
RED	#ff0000	FF 00 00	00 00 FF
GREEN	#00ff00	00 FF 00	00 FF 00
BLUE	#0000ff	00 00 FF	FF 00 00
WHITE	#ffffff	FF FF FF	FF FF FF
Favourite (previous assignment)	#00ffff	00 FF FF	FF FF 00

Screenshot modified BMP file in hex editor:



Assignment 1.4: Student number to HEX and Binary

Assignment

Convert your student number to a hexadecimal number and a binary number.
Explain in detail that the calculation is correct. Use the PowerPoint slides of week 1.
Convert your student number to a hexadecimal number and a binary number.
Explain in detail that the calculation is correct. Use the PowerPoint slides of week 1.

Answer

Student number: **569091**

Decimal to Binary conversion	
Divided by 2 rounded	**remainder**
284545	1
142272	1
71136	0
35568	0
17784	0
8892	0
4446	0
2223	0
1111	1
555	1
277	1
138	1
69	0
34	1
17	0
8	1
4	0
2	0
1	0
0	1
20 total values	1000 1010 1111 0000 0011

Binary outcome: **1000.1010.1111.0000.0011**
Explanation: divide by 2 each time and take the remainder.

Binary to Hexadecimal conversion		
Nibble (binary)	Decimal value	Hex digit
1000	8	8
1010	10	A
1111	15	F
0000	0	0
0011	3	3
1000 1010 1111 0000 0011		8AF03

Hex outcome: **0x8AF03**
Explanation: multiply each bit by its position weight then add them up to receive the decimal value. Then map the decimals to their corresponding hex value. See example table below for position weight.

Nibble position weight	
Bit position	3 2 1 0 (rightmost = 2^0 = 1)
Weights	8 × 4 × 2 × 1
1000 nibble example	1×8 + 0×4 + 0×2 + 0×1 = 8 + 0 + 0 + 0 = 8