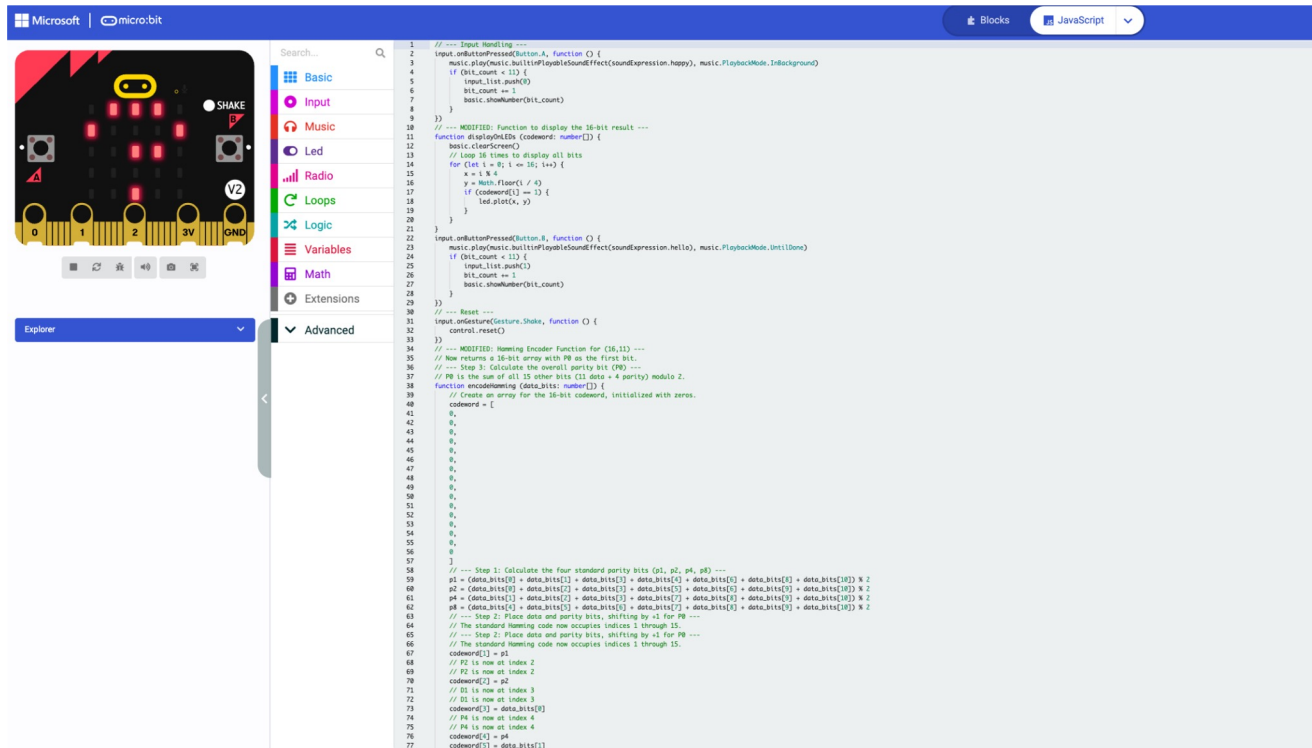


ENR-325/325L Principles of Digital Electronics and Laboratory

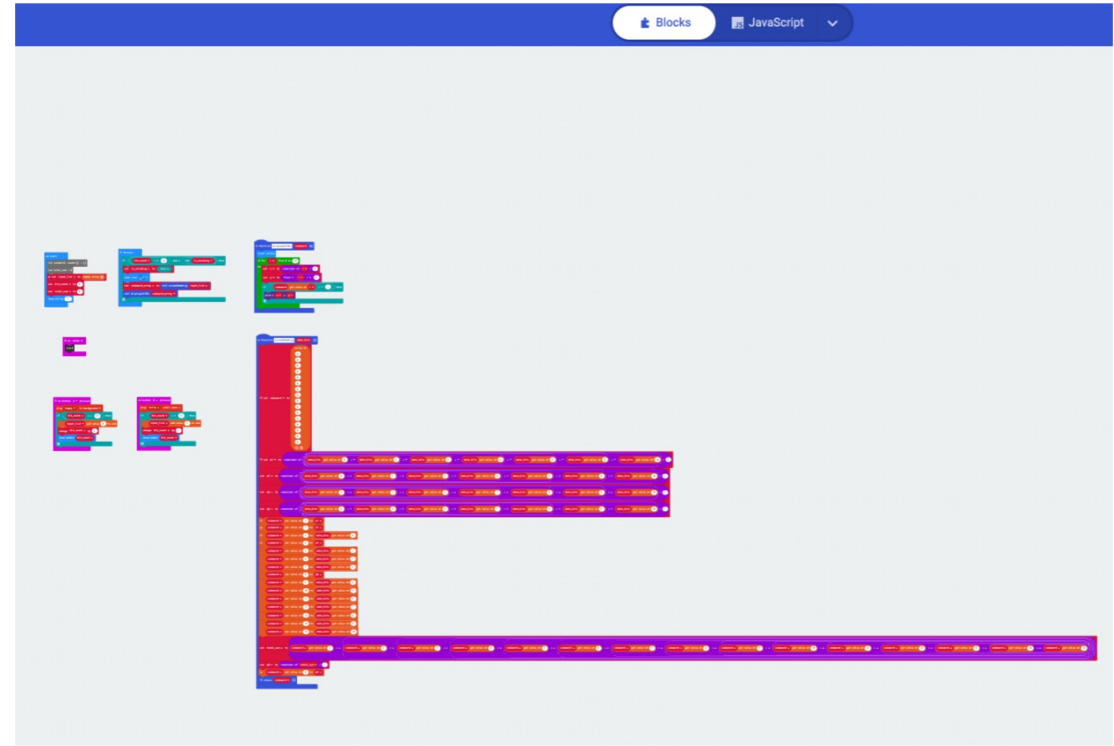
Xiang Li
Fall 2025

Hamming codes can be done in the CS way



The screenshot shows the Microsoft MakeCode editor interface for a micro:bit. On the left is a visual representation of the micro:bit board. The main area displays JavaScript code for a Hamming encoder. The code includes functions for input handling, displaying the 16-bit result, and encoding the data using a Hamming code. The code is as follows:

```
1 // --- Input Handling ---
2 input.onButtonPressed(Button.A, function () {
3   music.playMusic(builtInPlaybleSoundEffect(soundExpression.happy), music.PlaybackMode.InBackground)
4   if (bit_count < 12) {
5     input_list.push(0)
6     bit_count += 1
7     basic.showNumber(bit_count)
8   }
9 }
10
11 // --- MODIFIED: Function to display the 16-bit result ---
12 function displayLEDs (codeWord: number[]) {
13   basic.clearScreen()
14   // Loop 16 times to display all bits
15   for (let i = 0; i < 16; i++) {
16     x = i % 4
17     y = Math.floor(i / 4)
18     if (codeWord[i] == 1) {
19       led.plot(x, y)
20     }
21   }
22 }
23
24 input.onButtonPressed(Button.B, function () {
25   music.playMusic(builtInPlaybleSoundEffect(soundExpression.hello), music.PlaybackMode.UntilDone)
26   if (bit_count < 12) {
27     input_list.push(0)
28     bit_count += 1
29     basic.showNumber(bit_count)
30   }
31 }
32
33 // --- Reset ---
34 input.onGesture(Gesture.Shake, function () {
35   control.reset()
36 })
37
38 // --- MODIFIED: Hamming Encoder Function for (16,12) ---
39 // --- Now returns a 16-bit array with P0 as the first bit.
40 // --- Step 3: Calculate the overall parity bit (P0) ---
41 // P0 is the sum of all 15 other bits (11 data + 4 parity) modulo 2.
42 function encodeHamming (data: number[]) {
43   // Create an array for the 16-bit codeWord, initialized with zeros.
44   codeWord = [
45     0,
46     0,
47     0,
48     0,
49     0,
50     0,
51     0,
52     0,
53     0,
54     0,
55     0,
56     0,
57     0,
58     0,
59     0
60   ]
61   // --- Step 1: Calculate the four standard parity bits (p1, p2, p4, p8) ---
62   p1 = (data_bits[0] + data_bits[1] + data_bits[2] + data_bits[3] + data_bits[4] + data_bits[5] + data_bits[6] + data_bits[7]) % 2
63   p2 = (data_bits[0] + data_bits[2] + data_bits[4] + data_bits[6] + data_bits[8] + data_bits[10] + data_bits[12] + data_bits[14]) % 2
64   p4 = (data_bits[0] + data_bits[4] + data_bits[8] + data_bits[12] + data_bits[16] + data_bits[20] + data_bits[24] + data_bits[28]) % 2
65   p8 = (data_bits[0] + data_bits[8] + data_bits[16] + data_bits[24] + data_bits[32] + data_bits[40] + data_bits[48] + data_bits[56]) % 2
66   // --- Step 2: Place data and parity bits, shifting by +1 for P0 ---
67   // The standard Hamming code now occupies indices 2 through 15.
68   codeWord[2] = p1
69   // P2 is now at index 2
70   codeWord[3] = p2
71   // P4 is now at index 3
72   codeWord[4] = p4
73   // P8 is now at index 4
74   codeWord[5] = p8
75   // P4 is now at index 4
76   codeWord[6] = p4
77   codeWord[7] = data_bits[1]
```



Hamming codes can be done in the EE way

- Before that, we need to acquire some basic skill sets.

Pre-step: Data forms

Step 1: Data manipulation

Step 2: Information storage

Step 3: Interface

Pre-step: Data forms

- Say bye-bye to base 10:

Base 10 (0,1,2,3,4,5,6,7,8,9):

$$(4321)_{10} = 4 \times 10^3 + 3 \times 10^2 + 2 \times 10^1 + 1 \times 10^0$$

Base 2 (0,1):

$$(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

Base 16 (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F):

$$(FF12)_{16} = 15 \times 16^3 + 15 \times 16^2 + 1 \times 16^1 + 2 \times 16^0$$

Looking up how we do base conversions manually and in python.



COE COLLEGE®

The calculation of base 2 are pretty boring compared to base 10

Base 10

$$\begin{array}{r} 324 \\ +123 \\ \hline \end{array}$$

$$\begin{array}{r} 324 \\ -123 \\ \hline \end{array}$$

Base 2

$$\begin{array}{r} 110 \\ +101 \\ \hline \end{array}$$

$$\begin{array}{r} 110 \\ -101 \\ \hline \end{array}$$

Base 10

$$\begin{array}{r} 324 \\ \times 123 \\ \hline \end{array}$$

$$\begin{array}{r} 324 \\ \div 6 \\ \hline \end{array}$$

Base 2

$$\begin{array}{r} 110 \\ \times 101 \\ \hline \end{array}$$

$$\begin{array}{r} 110 \\ \div 10 \\ \hline \end{array}$$

We will revisit more binary arithmetic operation later, after the logic gates!

Discuss: the origin of base 16?

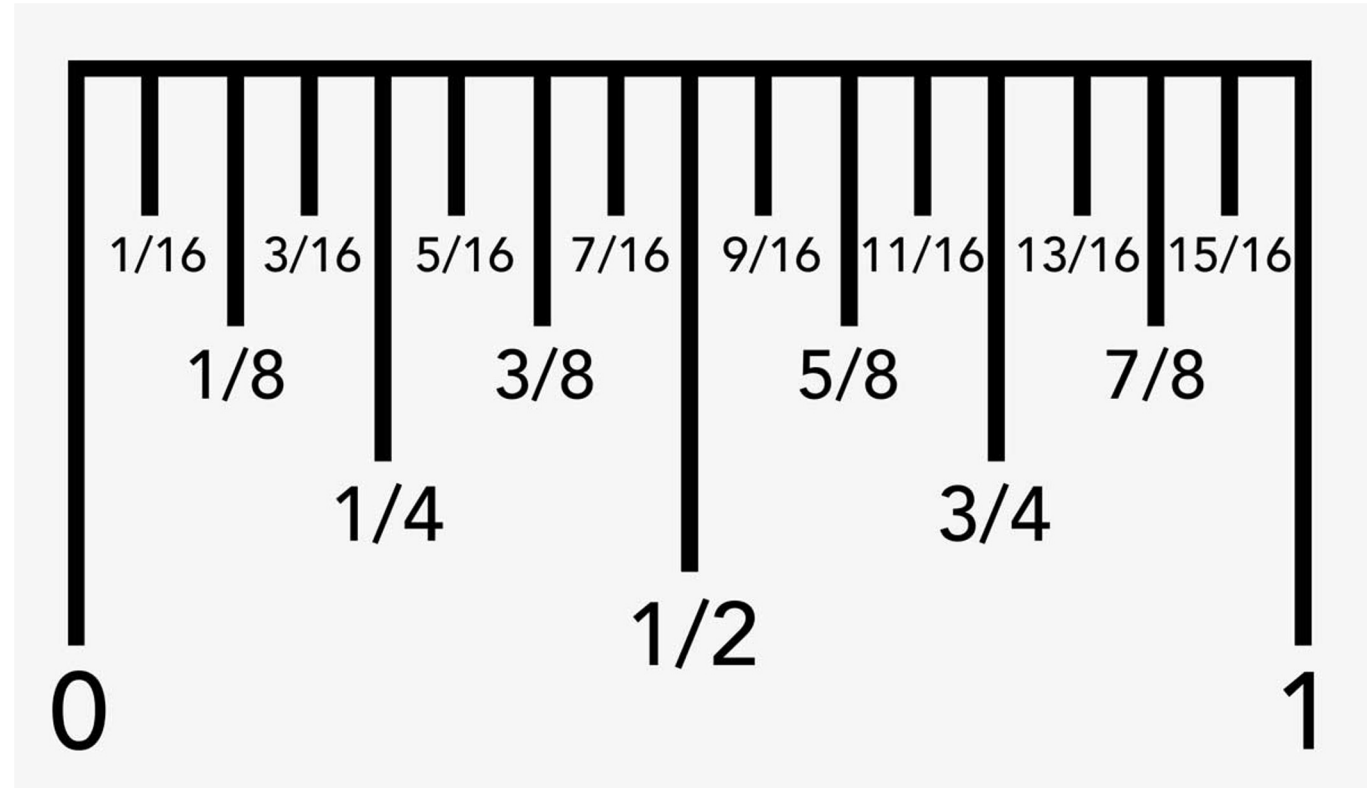
Discuss: the origin of base 16?

My theory:

An easy and fair way to compute with a weightless balance scale.



<https://commons.wikimedia.org/w/index.php?curid=79229218>



<https://www.inchcalculator.com/how-to-read-a-ruler/>



COE COLLEGE®

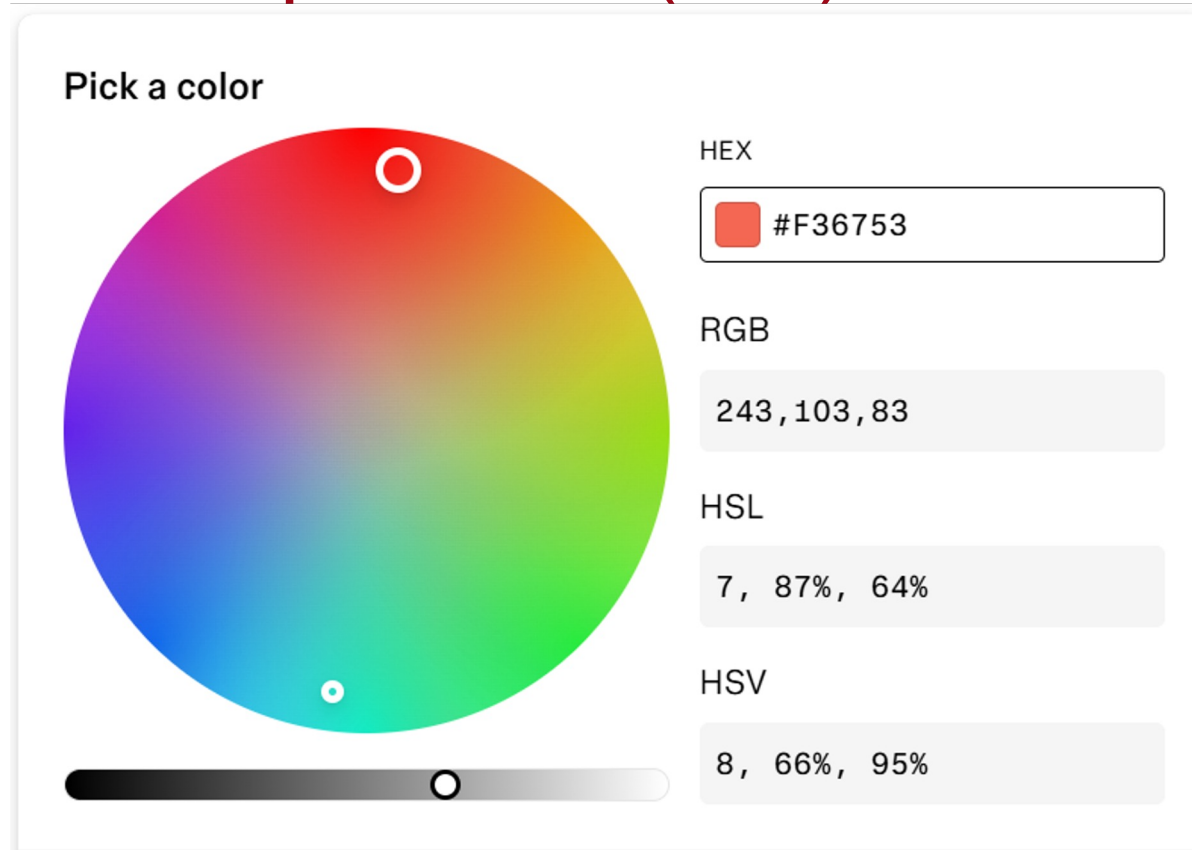
Discuss: why CS loves Hex(decimal) coding

0b : 0011100100101111010

0x : 392FA

Example: why CS loves Hex(decimal) coding

- Example: RGB (8bit) color code or Hex code



#F36753

R:
Bin(243)=11110011

G:
Bin(103)=01100111

B: Bin(83)=01010011

