

Numbering Systems

Our 10-fingered system

We use a numbering system based on 10 digits, 0 thru 9. *Decimal*. Dec = 10.

e.g. **1624** is

$$1 \times 10^3 + 6 \times 10^2 + 2 \times 10^1 + 4 \times 10^0 =$$

$$1 \times 1000 + 6 \times 100 + 2 \times 10 + 4 \times 1 =$$

$$1000 + 600 + 20 + 4 = \mathbf{1624}$$

Let's try another

1,267,442 is

$$1 \times 10^6 + 2 \times 10^5 + 6 \times 10^4 + 7 \times 10^3 + 4 \times 10^2 \\ + 4 \times 10^1 + 2 \times 10^0 =$$

$$(1 \times 1,000,000) + (2 \times 100,000) + \\ (6 \times 10,000) + (7 \times 1,000) + (4 \times 100) \\ + (4 \times 10) + (2 \times 1) = 1,267,442$$

Decimal System

- Base 10
- Number are written: 1624_{10}
- Uses digits 0 thru 9
- *Digits are one less than the base*

Works for any number “base”

- Base 8 (Octal) uses digits 0 – 7
- Base 4 uses digits 0 – 3
- Base 2 (Binary) uses digits 0 – 1

- Base 16 (Hexadecimal) uses “digits”:
0 1 2 3 4 5 6 7 8 9 A B C D E F

Base 8 (Octal)

749 - illegal

$$164_8 = 1 \times 8^2 + 6 \times 8^1 + 4 \times 8^0$$
$$= 64_{10} + 48_{10} + 4_{10} = 116_{10}$$

Base 2 (Binary)

14652 - illegal

$$10110101 = 1x2^7 + 0x2^6 + 1x2^5 + 1x2^4 \\ + 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0 =$$

$$128 + 0 + 32 + 16 + 0 + 4 + \\ 0 + 1 = 181_{10}$$

Base 16 - Hexadecimal

- Harder, because it's > 10, must have "digits" for 10, 11, 12, 13, 14, 15

$$\begin{aligned} \bullet A_{16} = 10_{10} \quad B_{16} = 11_{10} \quad C_{16} = 12_{10} \quad D_{16} = 13_{10} \\ E_{16} = 14_{10} \quad F_{16} = 15_{10} \end{aligned}$$

$$1AE2_{16} = 1 \times 16^3 + 10 \times 16^2 + 14 \times 16^1 \\ + 2 \times 1$$

$$= 4096 + 2560 + 224 + 2 \\ = 6882_{10}$$

Binary holds a special place

- 0, 1 called Binary Digits, or "bits"
- Eight binary digits:
from
0000 0000 (0_{10}) to 1111 1111 (255_{10})

called a "byte"

A Byte

- Eight bits
- Can represent a number from 0 to 255_{10}
- Reflect computer memory as a huge bank of 8-bit switches

- Switch is ON = 1
- Switch is OFF = 0

Extending bytes

- Use 7 bits
111 1111 - represent 0 to 127_{10}

- Use the highest bit for "sign"
0 111 1111 127

1 111 1111 (-127)

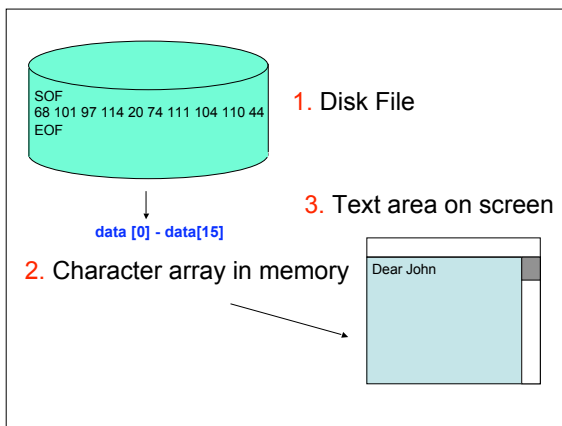
New Java type

```
public static byte x;
```

means that x can have a value from (-127)
to +127

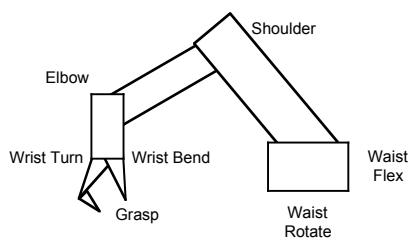
Computer Memory

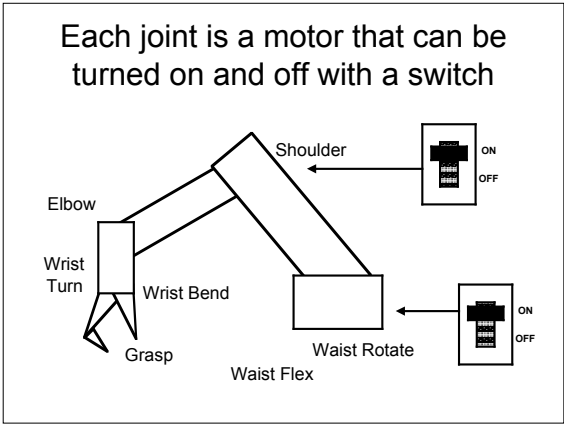
- “1 megabyte” of memory means one million, 8-bit (binary digit) numbers
- Each byte can hold a number from -127 to +127
- Called memory, because the computer uses each bit as a switch, that it sets to ON or OFF.

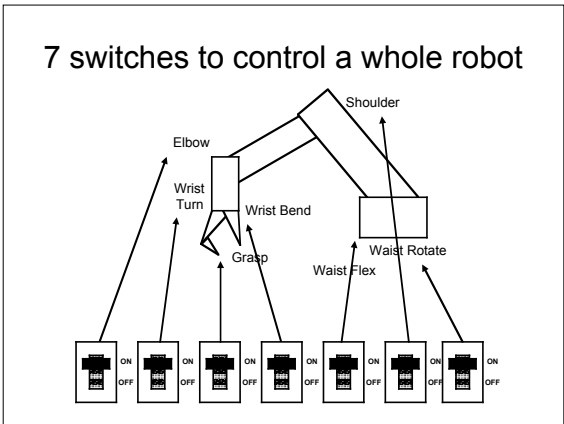


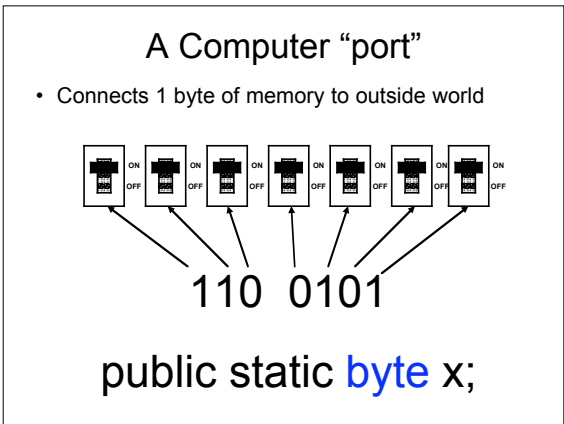
How do computers control the world?

A 7-jointed Robot









Bits translate into on/off

- $x = 127$ means 111 1111 means all motors or "ON"
- $x = 1$ means 000 0001 means the robot rotates at the waist
- $x = 16$ means 001 0000 means the robot grasps an object

A printer "Port"

- Computer connects an internal byte variable to the printer switches, and changes the values to tell the printer what to print

