

1. gyakorlat

Floating Point ← FPU

INT → egész szám
 ↳ 32 bit
 ↳ előjeles

Float → tört szám
 ↳ 32 bit
 ↳ előjeles

Char → ASCII

↳ egész számot tárol
 ↳ 8 bit
 ↳ előjeles

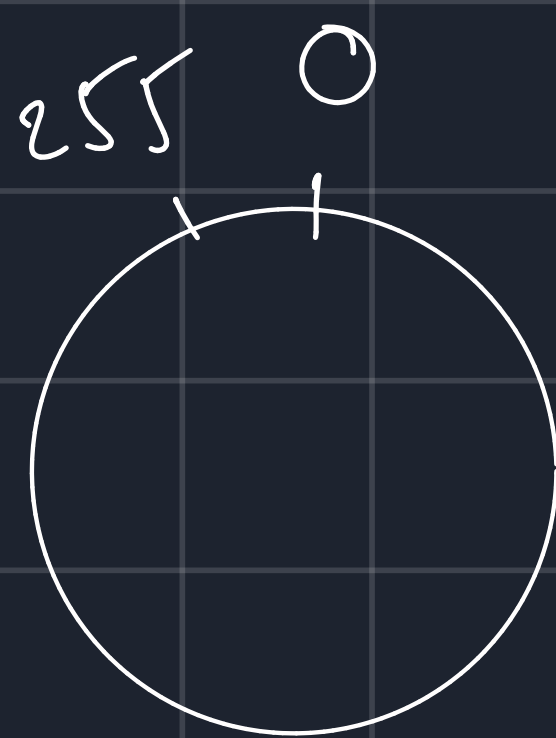
1 ZH - egész számok
 fixpontos sz.

2 ZH - lebegőpontos
 számok.

unsigned char

$$250 + 5 = 255$$

$$251 + 5 = 0$$



30 - 30 pont

1056

$$1 \cdot 10^3 + 0 \cdot 10^2 + 5 \cdot 10^1 + 6 \cdot 10^0$$

64 32 16 8 4 2 0

$$1010111 = 1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5 + 1 \cdot 2^6$$

fix pont (10-6)

610 | 1

305 | 1

152 | 0

76 | 0

38 | 0

19 | 1

9 | 1

4 | 0

2 | 0

1 | 1

100110011

tört
rész

eész rész

↳ 16 bites

8 bit | 8 bit

0,625

1 250

0 500

1 1000



$$610,625 = 1001100010,101$$

$$\begin{array}{r|l} 0 & 1 \\ \hline 0 & 2 \\ 0 & 4 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & 1 \end{array}$$

→ végtelen szorzás } Nem tudjuk a tizedestört $0,1$ -et eltárolni.

$$0.000110$$

$$\underbrace{1001100010,101}_{610}$$

$$\begin{array}{c} \downarrow \\ 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} \\ \underbrace{1 \cdot 2^{-1}}_{1/2=0,5} \quad \underbrace{0 \cdot 2^{-2}}_0 \quad \underbrace{1 \cdot 2^{-3}}_{1/8=0,125} \\ \hline 0,625 \end{array}$$

$$1056,123$$

$$\begin{array}{c} 1 \cdot 10^{-1} + 2 \cdot 10^{-2} + 3 \cdot 10^{-3} \\ 10^{-1} = \frac{1}{10^1} = \frac{1}{100} \end{array}$$

$$0.000110 \Rightarrow 0,09375 \rightarrow \text{mindig alulról közelítünk}$$