Declaring variables

char a; "signed 8-bit" -128 - 127

unsigned char b; "8-bit" 0 - 255

int i; "signed 16-bit" -32768 to 32767

unsigned int "16-bit" 0 - 65535

long "signed 32-bit"

-2,147,483,648

+2,147,483,647

unsigned long Tmr;

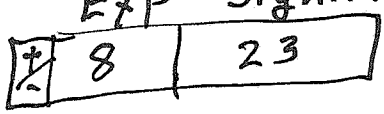
0 → 4,294,967,265

EtherNet

Media Access Control 48-bit 281.5 Trillion

64 bit 18.45 Quintillion

float x;



$\pm 2^{Exp-127} \cdot 1.\text{significand}$

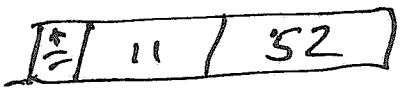
$0.9 \times 10^2 = 9 \times 10^1$

$1 \rightarrow 2$

$b_1 \quad b_{23}$

$1 + 2^{-1}b_1 + 2^{-2}b_2 + \ldots 2^{-23}b_{23}$

double



C operators

```
n = n+m;  // n += m;
n += 1; // n++;    ++n;
        // f(n++);   f(++n);
n = n-m; // n -= m;
n -= 1; // n--;  --n;
n = n*m; // n*= m;
n = n/m; // n /= m;
```

Bit wise Logicals

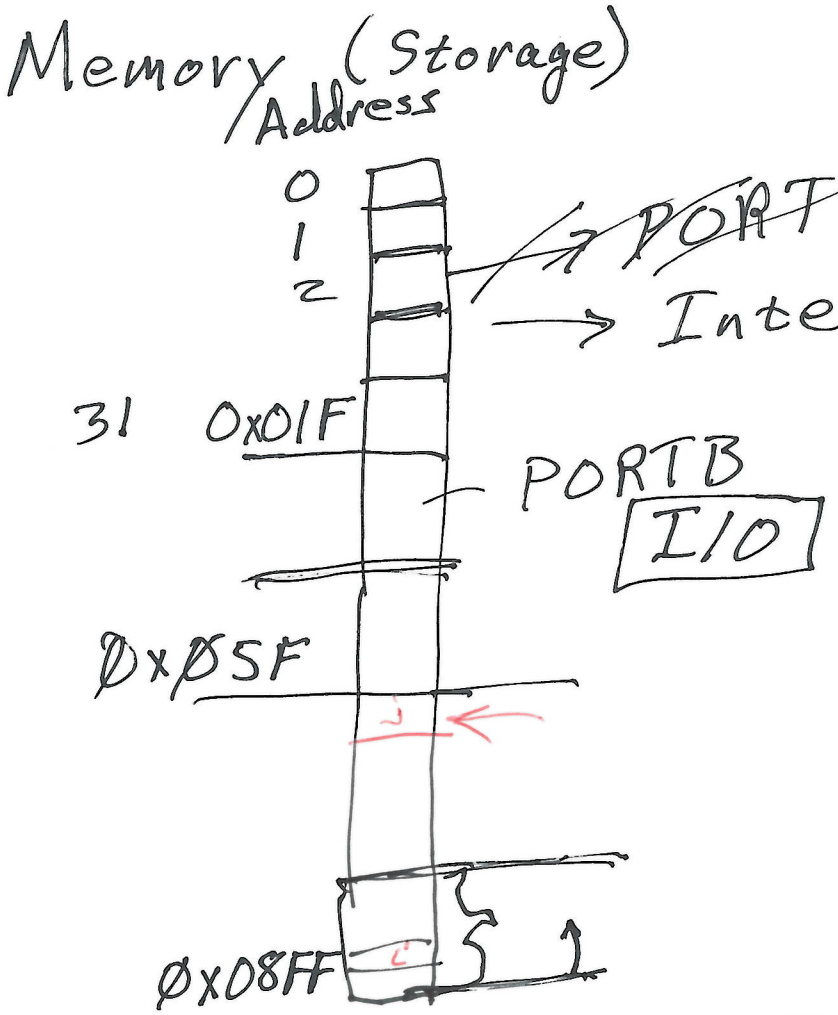
```
n &= mask;
n |= mask;
n ^= mask; // Only Arduino
n = n ^ mask;
n = m >> b;
```

n = xx1010 → xx1 , b = 3

```
bitRead(m, b)   (m >> b) & 1
bitSet(m,b) => m |= (1 << b)
```

0001
-01000

m | (1UL<<b)



---

unsigned long j;

```
unsigned long millis(void)
{ unsigned long i;

  return i;
}
```

i — Stack

```
char str[] = "This is a test";

str[3] = 's';

char str[32];
```