^

[Bitwise Operators]

Description

There is a somewhat unusual operator in C++ called bitwise EXCLUSIVE OR, also known as bitwise XOR. (In English this is usually pronounced "eks-or".) The bitwise XOR operator is written using the caret symbol ^. A bitwise XOR operation results in a 1 only if the input bits are different, else it results in a 0.

Precisely,

0 0 1 1 operand1

0 1 0 1 operand2

----------

0 1 1 0 (operand1 ^ operand2) - returned result

Example Code

int x = 12; // binary: 1100

int y = 10; // binary: 1010

int z = x ^ y; // binary: 0110, or decimal 6

The ^ operator is often used to toggle (i.e. change from 0 to 1, or 1 to 0) some of the bits in an integer expression. In a bitwise XOR operation if there is a 1 in the mask bit, that bit is inverted; if there is a 0, the bit is not inverted and stays the same.

// Note: This code uses registers specific to AVR microcontrollers (Uno, Nano, Leonardo, Mega, etc.)

// it will not compile for other architectures

void setup() {

DDRB = DDRB | B00100000; // set PB5 (pin 13 on Uno/Nano, pin 9 on Leonardo/Micro, pin 11 on Mega) as OUTPUT

Serial.begin(9600);

}

void loop() {

PORTB = PORTB ^ B00100000; // invert PB5, leave others untouched

delay(100);

}