

Darstellung von Daten:

Decimal:

```
int a = 5;                int a = 255;
```

Binär

```
int a = 0b00000101;      int a = 0b11111111;
```

Hexadezimal

```
int a = 0x05;            int a = 0xff;
```

Bitweise Operation (sehr sinnvoll für Mikrocontroller zum Beschreiben der Register)

Setzen des 0. Bits (ohne die anderen Bits in dem Byte PORTB zu verändern)

```
PORTB |= _BV(PB0);        // über ein Makro
PORTB = PORTB | _BV(PB0);  // langversion
PORTB |= (1<<0);           // durch Schiebeoperation (eine 1 um 0 mal nach Links)
PORTB |= 0x01;             // durch eine einfache Oder-Operation mit einer 1 (Hex)
PORTB |= 0b00000001;       // durch eine einfache Oder-Operation mit einer 1 (Binär)
| : bitweise ODER-Verknüpfung
```

Löschen des 0. Bits (ohne die anderen Bits in dem Byte PORTB zu verändern)

```
PORTB &= ~_BV(PB0);        // über ein Makro (gleicher Aufbau wie beim Setzen)
PORTB = PORTB & ~_BV(PB0); // langversion
& : bitweise UND-Verknüpfung
~ : bitweise Invertierung des dahinter stehenden Wertes
```

Invertieren des 0. Bits (ohne die anderen Bits in dem Byte PORTB zu verändern)

```
PORTB ^= _BV(PB0);         // über ein Makro (gleicher Aufbau wie beim Setzen)
PORTB = PORTB ^ _BV(PB0);   // langversion
^ : bitweise XOR-Verknüpfung
0b11001100 ^ 0b00000001 => 0b11001101
0b11001101 ^ 0b00000001 => 0b11001100
```

Nützliche Links (Mikrocontroller):

[1] <https://www.mikrocontroller.net/articles/AVR-GCC-Tutorial>

Nützliche Links (Programmiersprache C):

[10] Google ;)