ATmega128 Timers with interrupts in C

Set up tick like this (This is with a prescaler of 64, duh):

```
#define tick F_osc/64
```

OCR1A - Used like:

```
TIMSK = (1 << OCIE1A)
```

To send an interrupt when timer 1 matches the value of OCR1A (Set to the tick you want).
The interrupt vector for this is

```
ISR(TIMER1_COMPA_vect)
```

Use it as a function doing code that will happen when the compare matches.

If TCCR1B was initialized with (1 << WGM12), the timer will be reset when the comparison matches.
Initialize TCCR1B with prescaler and reset on match like this (example):

```
TCCR1B = (1 << CS10) | (1 << CS11) | (1 << WGM12)
```

Timer prescaler is set with CS10, CS11 and CS12 like this:

Table 62. Clock Select Bit Description

CSn2	CSn1	CSn0	Description
0	0	0	No clock source. (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/1 (No prescaling
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on Tn pin. Clock on falling edge
1	1	1	External clock source on Tn pin. Clock on rising edge

Figure 1: Prescaling

Set value of timer like this (set to zero before using):

```
TCNT1 = 0x0;
```

For other timers, replace 1 with the corresponding timer you want to use.

Bitwise operations in C

The AND (&) operator

bit a	bit b	a & b (a AND b)
0	0	0
0	1	0
1	0	0
1	1	1

Figure 2: Examples of result of &

The OR (|) operator

bit a	bit b	a | b (a OR b)
0	0	0
0	1	1
1	0	1
1	1	1

Figure 3: Examples of result of |

The NOT (~) operator

bit a	~a (complement of a)
0	1
1	0

Figure 4: Examples of result of ~

The XOR (^) operator

bit a	bit b	a ^ b (a XOR b)
0	0	0
0	1	1
1	0	1
1	1	0

Figure 5: Examples of result of ^

Bit-shifting

Using << (left-shift) will shift the current bits to the left.
Like this: 0b00001010 → 0b00010100

>> will do the opposite.
Like this: 0b00010100 → 0b00001010