

Init Assembly!

```
init:

.equ F_osc = 4000000 #klokkefrekvens
.equ tick = F_osc/1024 #klokkefrekvens med prescalar

#init stack
.def tmp = R16 #definer tmp til register 16

ldi tmp, low(ramend)
out spl, tmp
ldi tmp, high(ramend)
out sph, tmp

#init port
ldi tmp, 0x0 #clear tmp til 0 for å sette som output (low, alle bit til 0)
out DDRA, tmp
ldi tmp, 0xFF #sett tmp til 1 for å sette som input (high, alle bit til 1)
out DDRB, tmp
ldi tmp, 0x2 #sett bit 1 til 1 (high), kontroll bit satt til input
#(rest av bits er output)
out DDRC, tmp

#init timer

#timer compare
ldi tmp, low(tick)
out OCR1BL, tmp
ldi tmp, high(tick)
out OCR1BH, tmp

#timer counter
ldi tmp, (1 << WGM12) | (1 << CS12) | (1 << CS10) #WGM12 = CTC,
#CS12 + CS10 = 1024 prescaler
out TCCRIB, tmp

#clear timer counter
clr tmp
out TCNT1L, tmp
out TCNT1H, tmp

#enable interrupts on compare
ldi tmp, (1 << OCIE1B)
out TIMSK, tmp
```

```

#enable interrupts
sei

.org OCR1Baddr #adresse 18 - hopp til ISR
jmp ISR

ISR:
push tmp #push tmp til stack
in tmp,sreg #hent fra sreg til tmp
push tmp #push tmp til stack (sreg)
...
...
pop tmp #hent tmp fra stack
out sreg, tmp #send tmp til sreg
pop tmp #hent tmp fra stack
reti

```