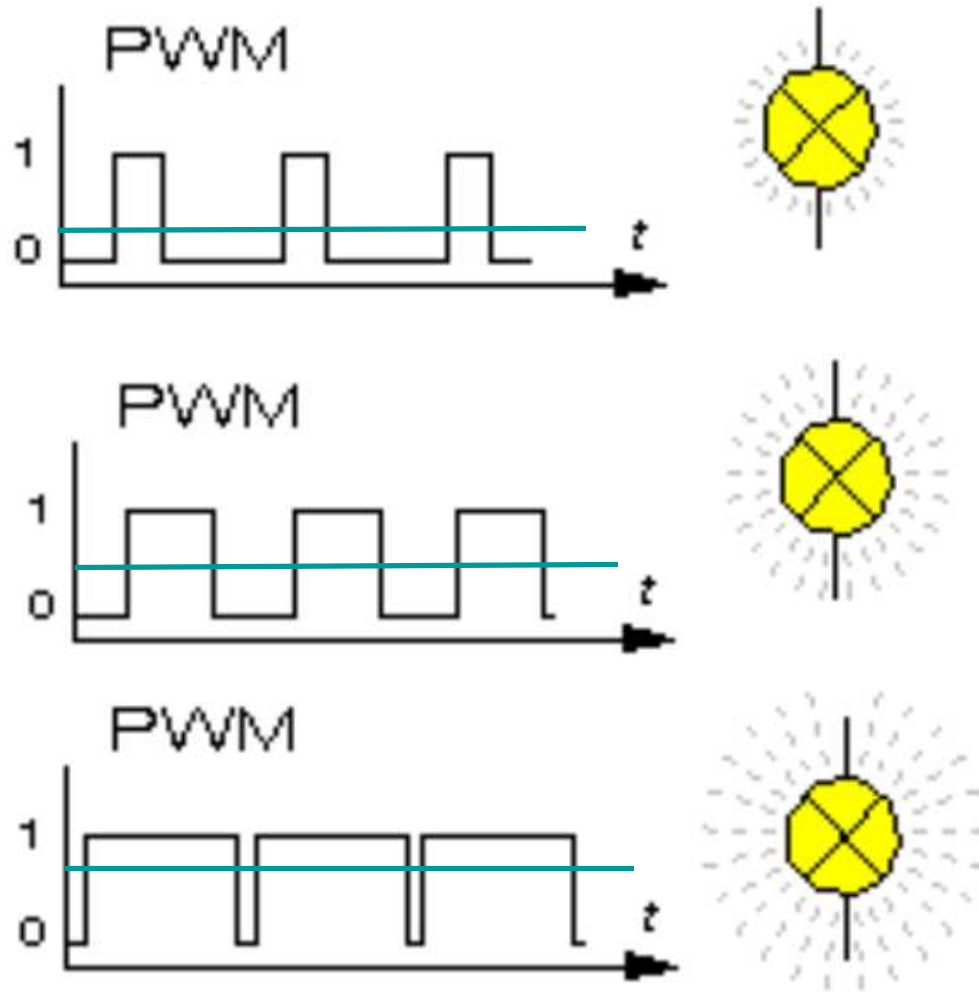
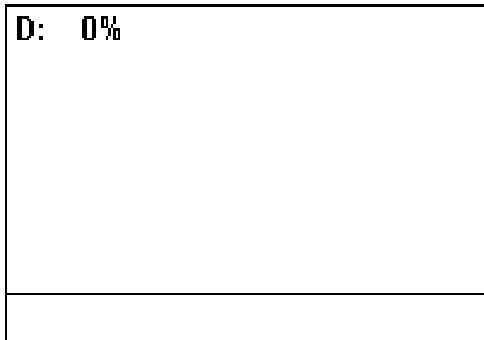


DA-omvandling, oftast PWM

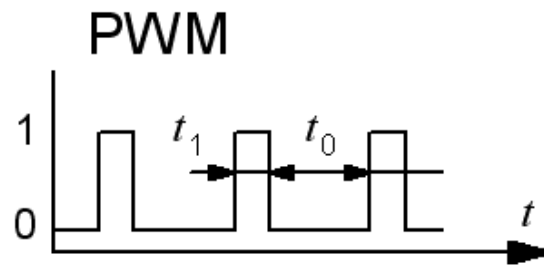
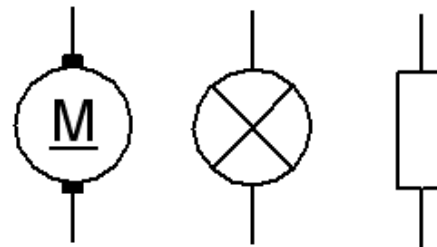


En DA-omvandlare tar stor plats på processor-chippet. Den vanligaste DA-lösningen är i stället en pulsbreddsmodulator. Många komponenter ”märker” ingen skillnad mellan ett stabilt analogt värde eller **medelvärde** från snabba pulser.

DA-omvandling, oftast PWM

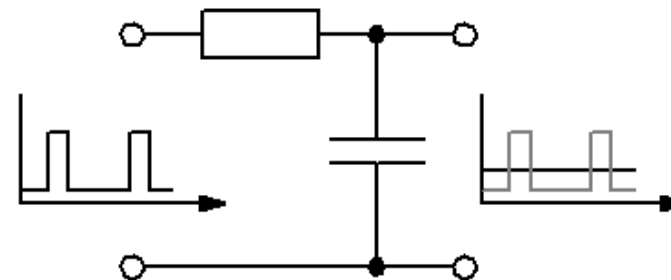


Tröga komponenter
= medelvärde

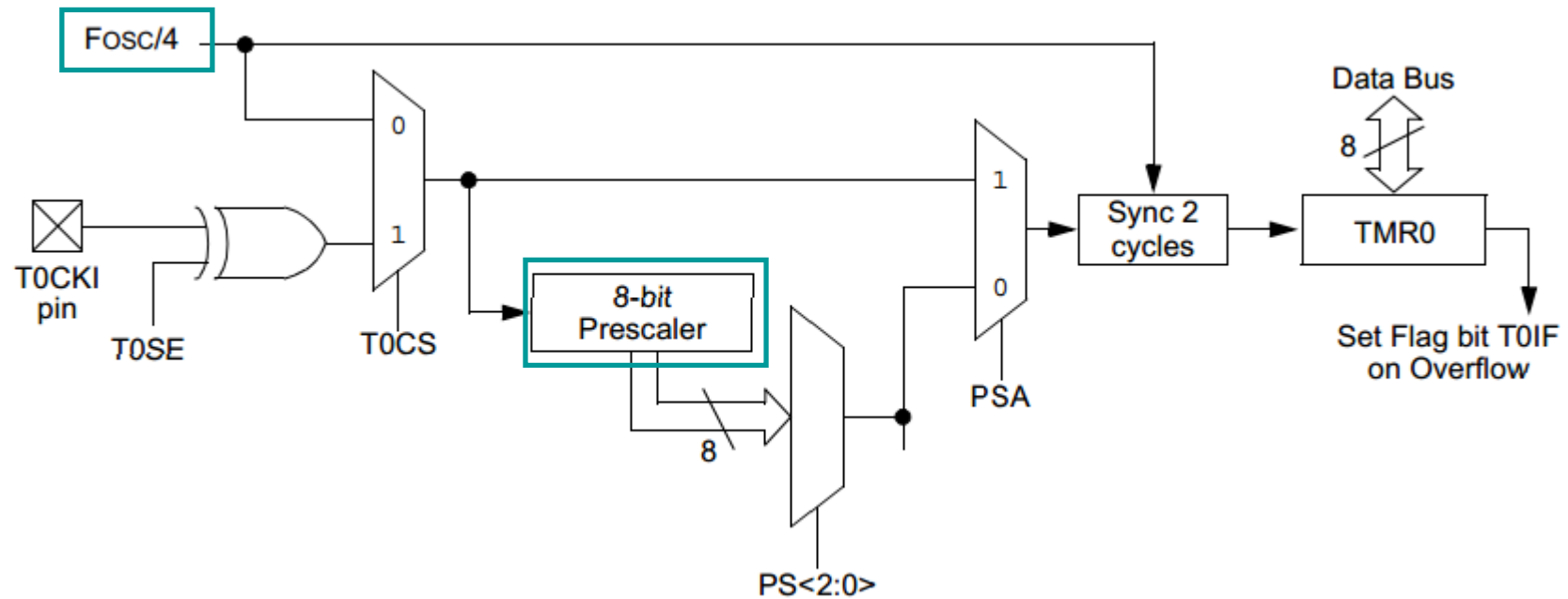


$$\text{DutyCycle} = \frac{t_1}{t_1 + t_0}$$

Lågpasfilter
= likspänning



TIMER0



REGISTER 5-1: OPTION_REG: OPTION REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RABPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

PWM-program

```
#define DUTY 128
```

```
void main( void)  
{
```

```
    TRISC.5 = 0;          /* PORTC.5 is output */
```

```
    OPTION = 0b10000.111; /* 256 prescale */
```

```
    while (1) /* forever */
```

```
    {
```

```
        char i;
```

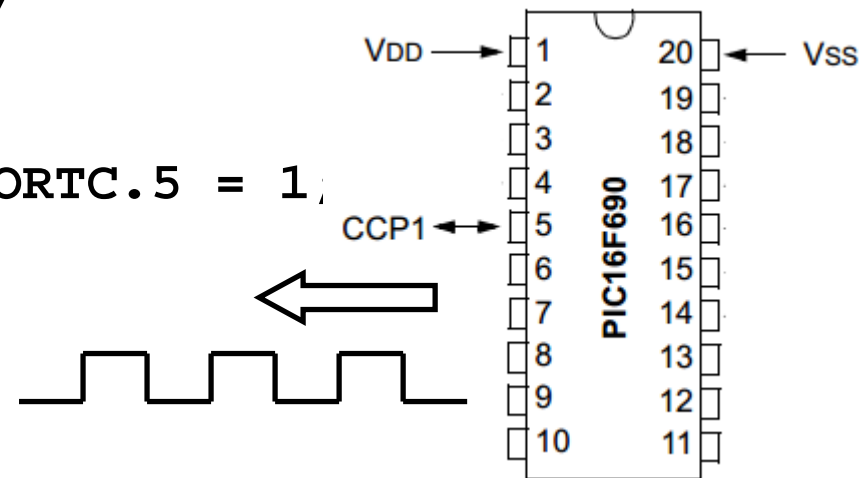
```
        if (TMR0 < DUTY ) PORTC.5 = 1;
```

```
        else PORTC.5 = 0;
```

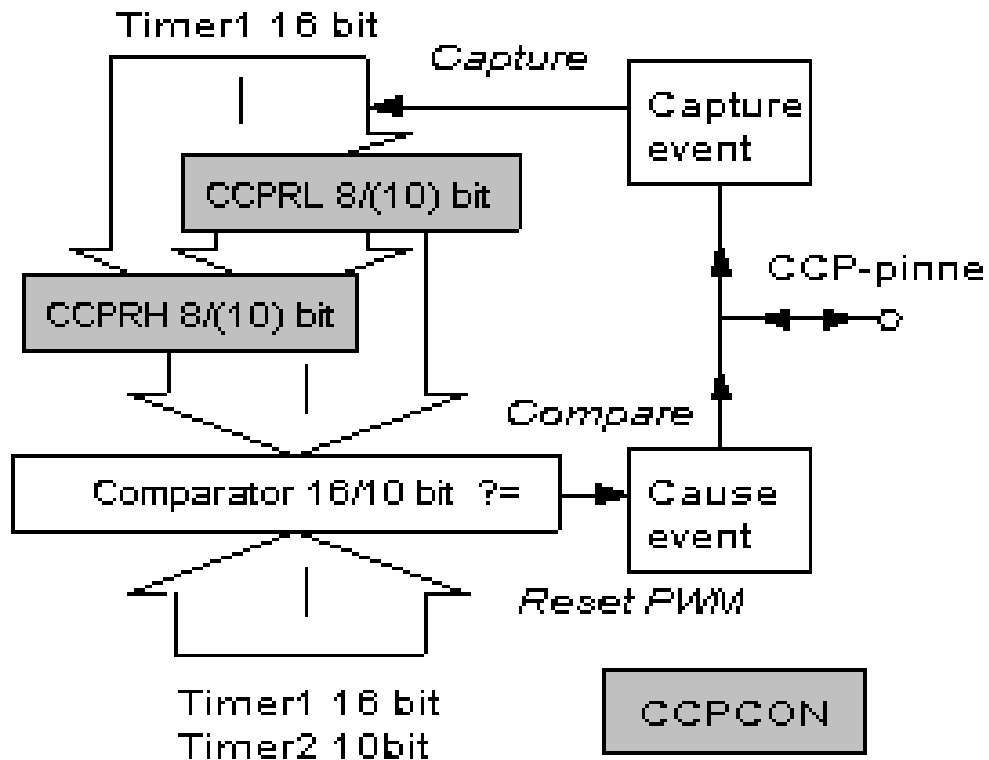
```
    }
```

```
}
```

Problem. Behöver programmet göra något mer så får det ske mellan TIMER0 tick'en!



CCP-enhet



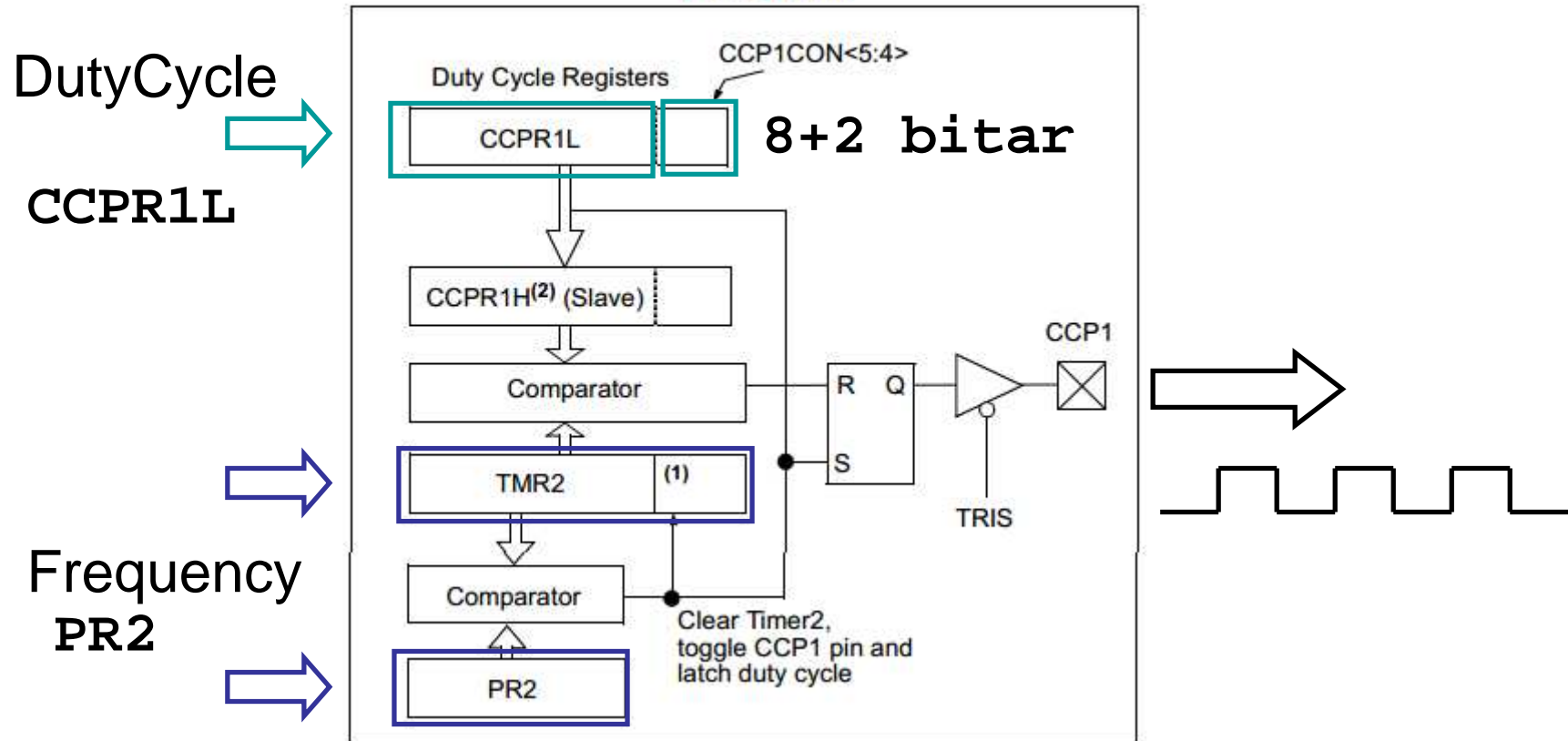
CCP

- Capture
- Compare
- PWM

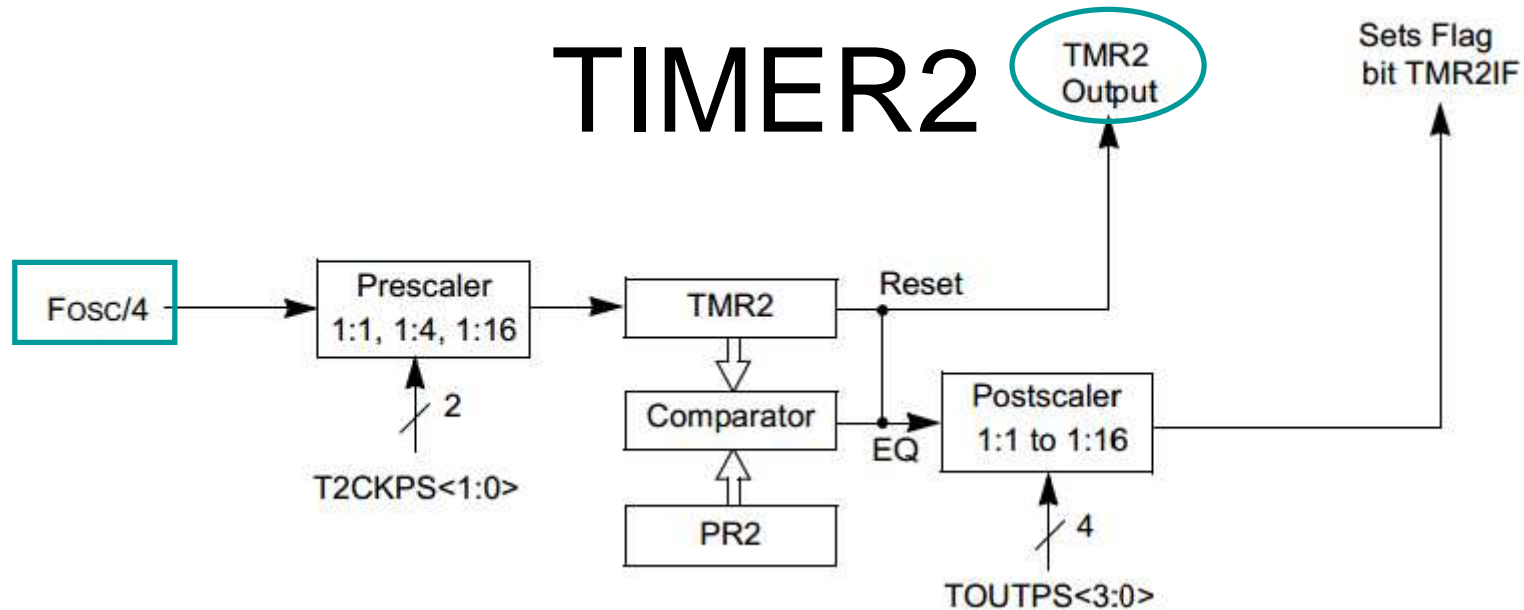
Det finns ett uppenbart behov av en *självgående enhet* för generering av PWM. **CCP-enheten** kan programmeras till detta!

PWM

FIGURE 11-3: SIMPLIFIED PWM BLOCK DIAGRAM



TIMER2



TIMER2 är en 8-bit räknare (upp till modulo 256). Den har en **prescaler** från processorklockan, och ett register **PR2** som kan ”förkorta” räknecykeln – den räknar då ”modulo PR2”. Detta ger många möjligheter till att ställa in TIMER2 Output frekvensen.

REGISTER 7-1: T2CON: TIMER 2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

PWM f 1kHz D 50%

Antag att vi behöver generera en PWM-signal med f **1 kHz** och dutycycle **50%** (någon som gillar jämna siffror).

$$\frac{f_{osc}}{4} = \frac{4 \cdot 10^6}{4} = 1 \cdot 10^6 \quad \frac{1 \cdot 10^6}{\text{prescale}\{1 \quad \boxed{4} \quad 16\}} = 250 \cdot 10^3$$

$$\text{PR2} = 249 \quad [0 \dots 249] \quad \frac{250 \cdot 10^3}{249 + 1} = \boxed{1000 \text{ Hz}}$$

REGISTER 7-1: T2CON: TIMER 2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0
	-	-	-	-	1	0	1

PWM f 1kHz D 50%

REGISTER 11-1: CCP1CON: ENHANCED CCP1 CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
bit 7							bit 0

0 0
CCP1-pin

1 1 0 0

PWM-mode

`TRISC.5=0;`

Två extra bitars DutyCycle
upplösning (minst signifikanta bitar)

Tio bitars upplösning har man när $PR2=255$. Lägre värde ger reducerad upplösningen enligt formeln:

$$\text{resolution} = \frac{\log(4 \cdot (PR2 + 1))}{\log(2)} \quad [\text{bits}]$$

PWM f 1kHz D 50%

REGISTER 11-1: CCP1CON: ENHANCED CCP1 CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
bit 7							bit 0

$$\text{DutyCycle} = \frac{\text{CCPR1L} \cdot 4 + \text{DC1B1} \cdot 2 + \text{DC1B0} \cdot 1}{(\text{PR2} + 1) \cdot 4}$$

$$50\% = \frac{\text{CCPR1L} \cdot 4 + 0 \cdot 2 + 0 \cdot 1}{(249 + 1) \cdot 4} \Rightarrow \text{CCPR1L} = \frac{0,5 \cdot 4 \cdot 250}{4} = 125$$

Ex. Vilken DutyCycle?

PR2 = 208;

CCPR1L = 137;

DC1B1 = 1;

DC1B0 = 1;

?

$$\text{DutyCycle} = \frac{\text{CCPR1L} \cdot 4 + \text{DC1B1} \cdot 2 + \text{DC1B0} \cdot 1}{(\text{PR2} + 1) \cdot 4} =$$

$$= \frac{137 \cdot 4 + 1 \cdot 2 + 1 \cdot 1}{(208 + 1) \cdot 4} = 65,9\%$$

Ex. Vilken DutyCycle?

PR2 = 208;

CCPR1L = 209;

DC1B1 = 1;

DC1B0 = 1;

?

$$\text{DutyCycle} = \frac{\text{CCPR1L} \cdot 4 + \text{DC1B1} \cdot 2 + \text{DC1B0} \cdot 1}{(\text{PR2} + 1) \cdot 4} =$$

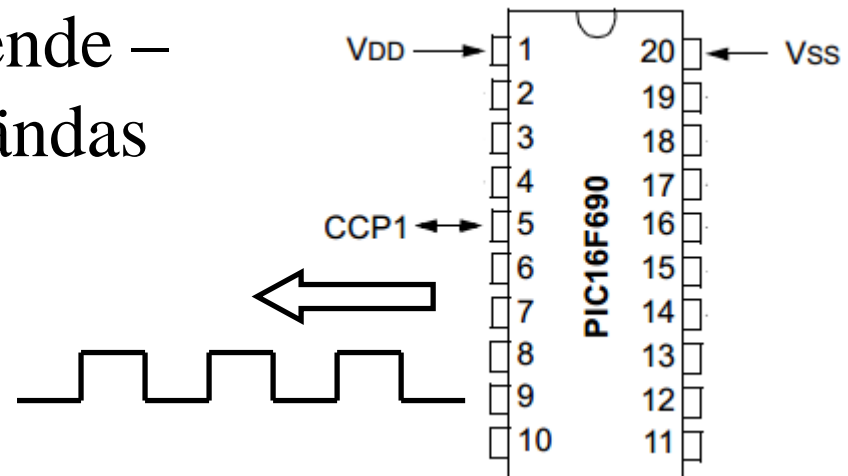
$$= \frac{209 \cdot 4 + 1 \cdot 2 + 1 \cdot 1}{(208 + 1) \cdot 4} = 100,4\%$$

Fungerar inte!
CCPR1L får aldrig
vara större än **PR2**!

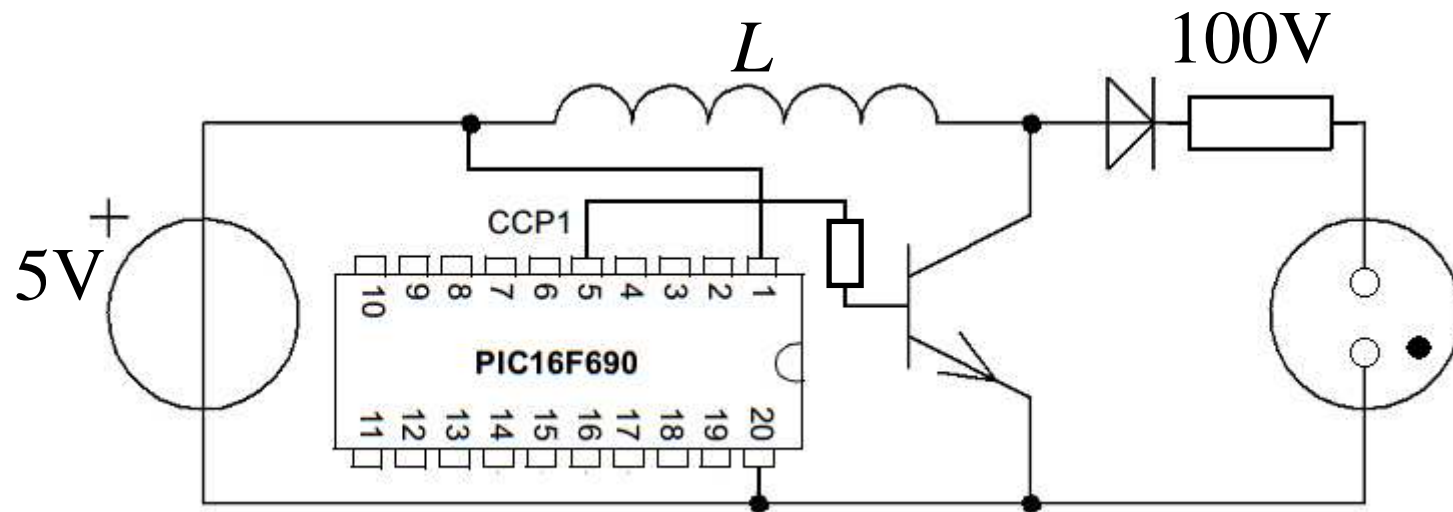
PWM f 1kHz D 50%

```
TRISC.5=0;          /* CCP1 output    */
T2CON = 0B00000101; /* prescale 1:4  */
CCP1CON = 0B00.00.1100; /* PWM-mode     */
PR2 = 249;         /* f_pwm 1000 Hz */
CCPR1L = 125;     /* Duty 50%     */
```

CCP-enheten är självgående –
all processortid kan användas
till annat.

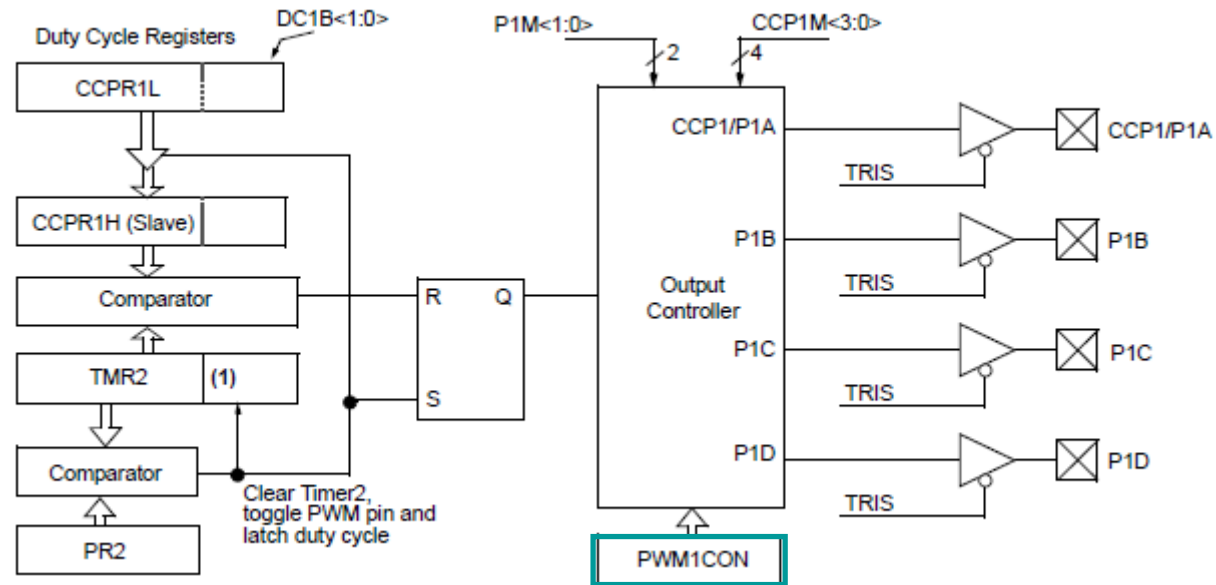


PWM till StepUp



Vid laborationen kommer Du att behöva 100V för att tända en glimlampa!

PWM till motorer



Ett annat vanligt användningsområde för PWM är till motorstyrning – vi återkommer till det.



William Sandqvist william@kth.se