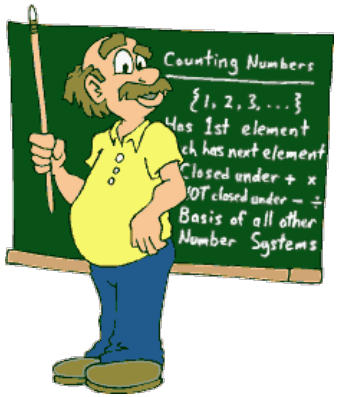
AARHUS
UNIVERSITY
SCHOOL OF ENGINEERING

MSYS

Microcontroller Systems

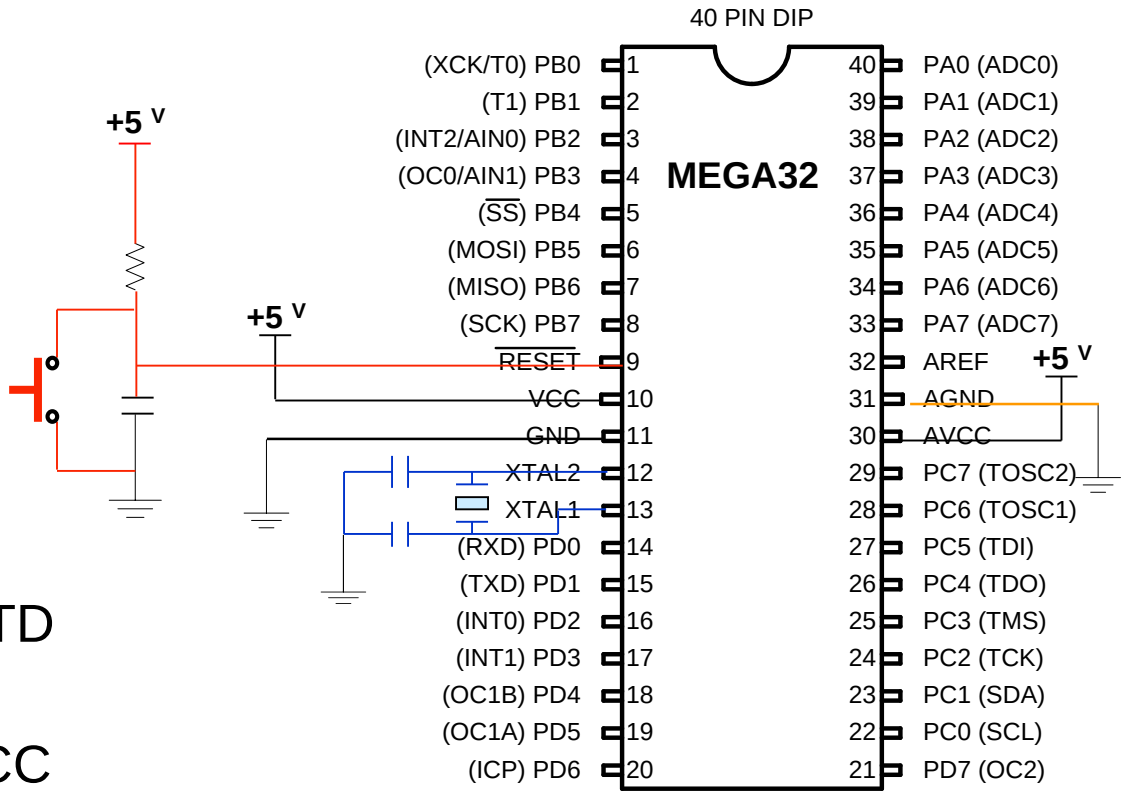
Lektion 8: Digitale Porte



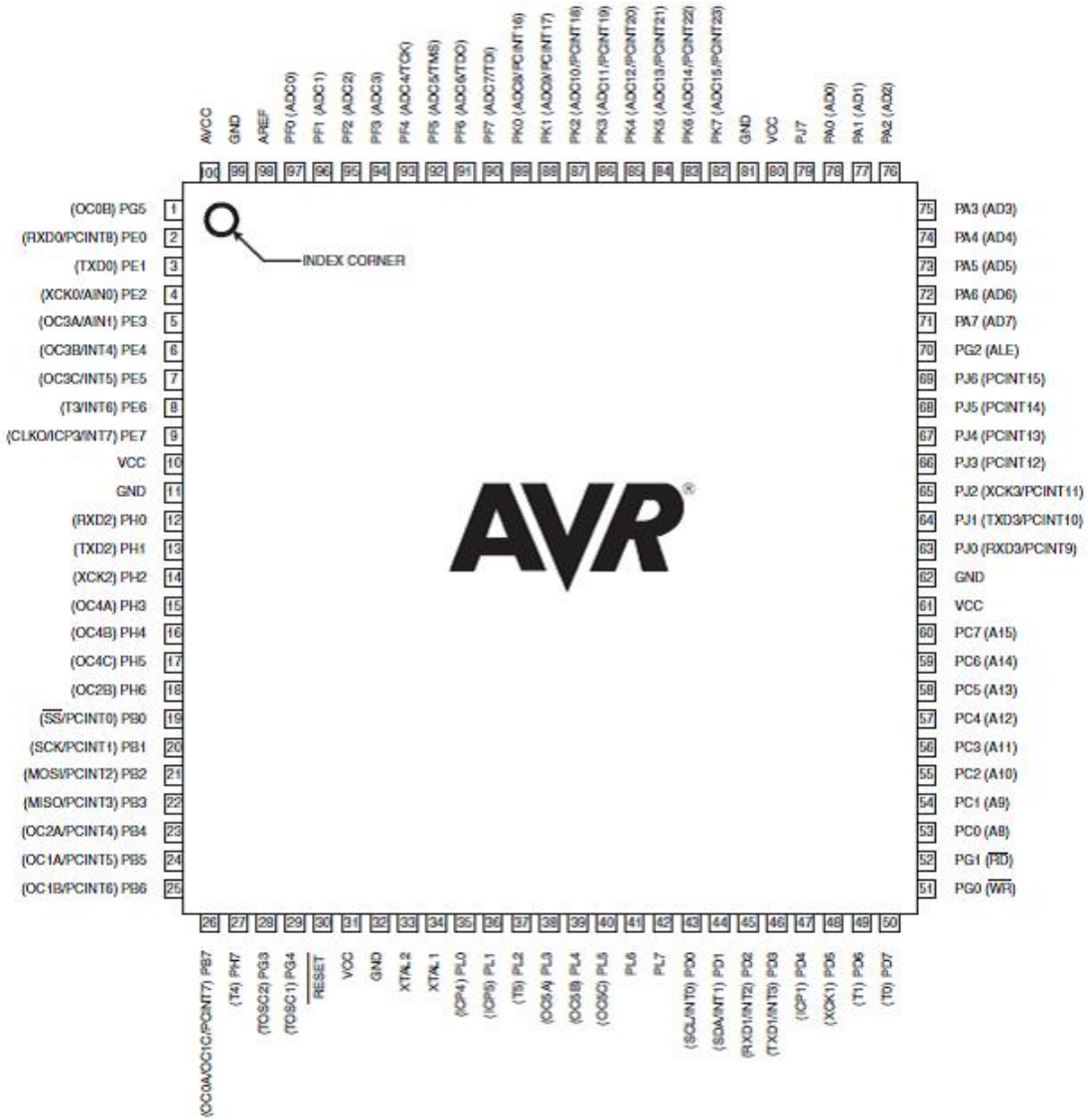
Version 12-9-2017, Henning Hargaard

Mega32 Pinout

1. Vigtige ben:
   1. Power
      - VCC
      - Ground
   2. Clock
      - XTAL1
      - XTAL2
   3. Reset
2. Digital I/O
   - PORTA, PORTB, PORTC, and PORTD
3. ADC pins
   - AREF, AGND, AVCC



Mega2560 Pinout



Forskellige AVR controllers Porte

Mega32

Mega2560

Table 4-1: Number of Ports in Some AV nbers

Pins	8-pin	28-pin	40-pin	64-pin	100-pin
Chip	ATtiny25/45/85	ATmega8/48/88	ATmega32/16	ATmega64/128	ATmega1280
Port A			X	X	X
Port B	6 bits	X	X	X	X
Port C		7 bits	X	X	X
Port D		X	X	X	X
Port E				X	X
Port F				X	X
Port G				5 bits	6 bits
Port H					X
Port J					X
Port K					X
Port L					X

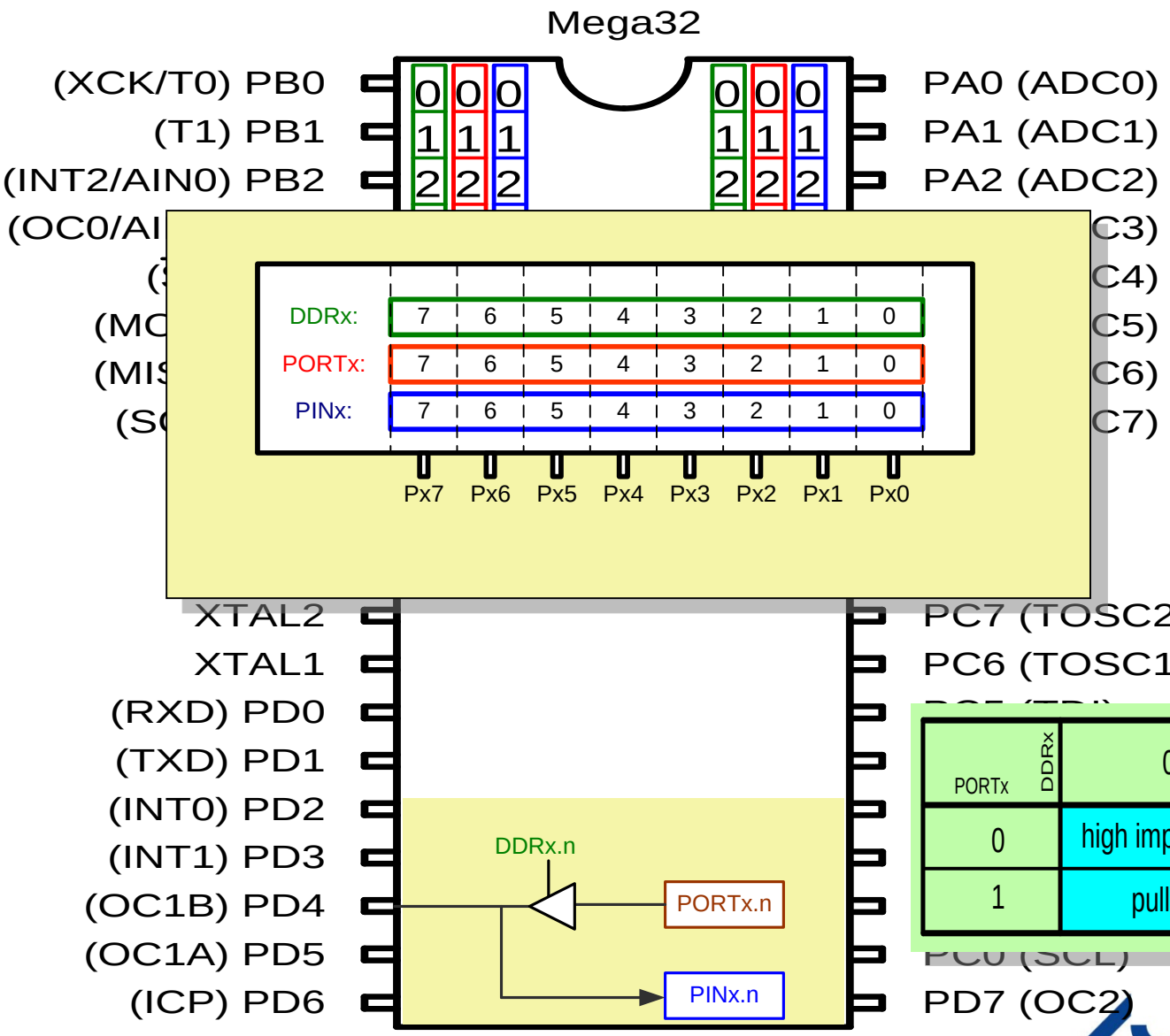
Note: X indicates that the port is available.

32 I/O ben

86 I/O ben

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

Digitale porte og registre



PORTx \ DDRx	0	1
0	high impedance	Out 0
1	pull-up	Out 1

Test (”socrative.com”: Room = MSYS)

- Vi ønsker, at alle ben på PD skal være udgange. Hvordan gøres det ?

A:
```
CLR R16
OUT DDRD,R16
```

B:
```
CLR R16
OUT PORTD,R16
```

C:
```
SER R16
OUT DDRD,R16
```

D:
```
SER R16
OUT PIND,R16
```

I/O adresser for porte i Mega32

”IN eller OUT”

	$1B ($3B)	PORTA
	$1A ($3A)	DDRA
	$19 ($39)	PINA
	$18 ($38)	PORTB
	$17 ($37)	DDRB
	$16 ($36)	PINB
	$15 ($35)	PORTC
	$14 ($34)	DDRC
	$13 ($33)	PINC
	$12 ($32)	PORTD
	$11 ($31)	DDRD
	$10 ($30)	PIND

($ = 0x = hex)

I/O adresser for porte i Mega2560

”IN eller OUT”

0x0B (0x2B)	PORTD
0x0A (0x2A)	DDRD
0x09 (0x29)	PIND
0x08 (0x28)	PORTC
0x07 (0x27)	DDRC
0x06 (0x26)	PINC
0x05 (0x25)	PORTB
0x04 (0x24)	DDRB
0x03 (0x23)	PINB
0x02 (0x22)	PORTA
0x01 (0x21)	DDRA
0x00 (0x20)	PINA

”IN eller OUT”

0x14 (0x34)	PORTG
0x13 (0x33)	DDRG
0x12 (0x32)	PING
0x11 (0x31)	PORTF
0x10 (0x30)	DDRF
0x0F (0x2F)	PINF
0x0E (0x2E)	PORTE
0x0D (0x2D)	DDRE
0x0C (0x2C)	PINE

I/O adresser for porte i Mega2560

”LDS eller STS”

(0x10B)	PORTL
(0x10A)	DDRL
(0x109)	PINL
(0x108)	PORTK
(0x107)	DDRK
(0x106)	PINK
(0x105)	PORTJ
(0x104)	DDRJ
(0x103)	PINJ
(0x102)	PORTH
(0x101)	DDRH
(0x100)	PINH

OBS:

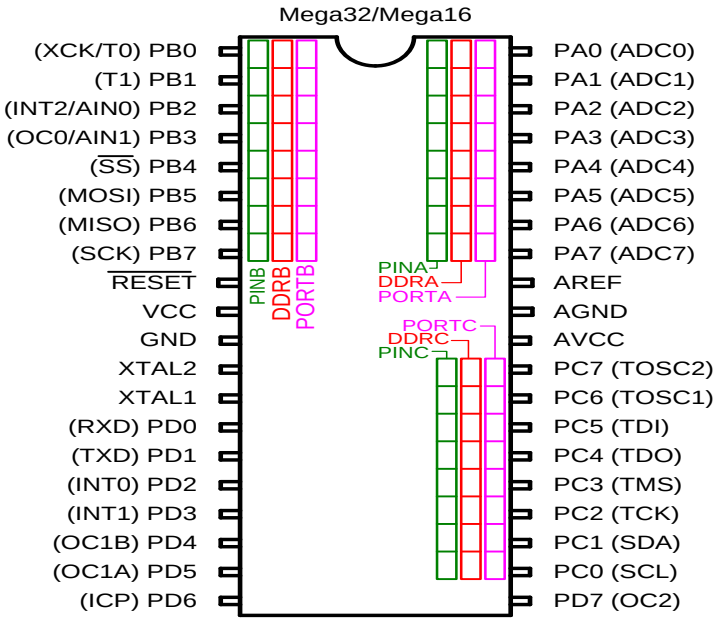
For portene
H, J, K og L
Kan man **IKKE**
anvende **IN- eller**
OUT instruktioner

Example 1

- Write a program that makes all the pins of PORTA one.

DDRA:	1	1	1	1	1	1	1	1
PORTA:	1	1	1	1	1	1	1	1

```
LDI   R20,0xFF   ;R20 = 11111111 (binary)
OUT   PORTA,R20  ;PORTA = R20
OUT   DDRA,R20   ;DDRA = R20
```



PORTx \ DDRx	0	1
0	high impedance	Out 0
1	pull-up	Out 1

Example 2

- The following code will toggle all 8 bits of Port B forever with some time delay between “on” and “off” states:

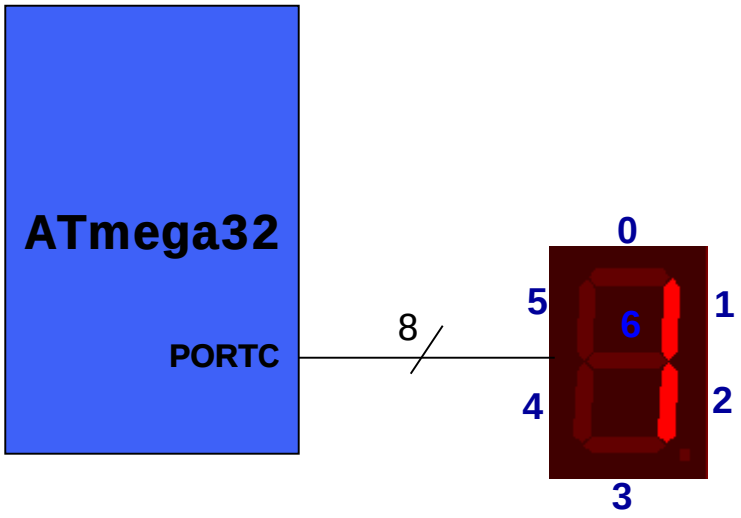
```
    LDI   R16,0xFF    ;R16 = 0xFF = 0b11111111
    OUT   DDRB,R16    ;make Port B an output port (1111 1111)
L1: LDI   R16,0b01010101
    OUT   PORTB,R16   ;put 01010101 on port B pins
    CALL   DELAY
    LDI   R16,0b10101010
    OUT   PORTB,R16   ;put 10101010 on port B pins
    CALL DELAY
    JMP L1
```

Example 3

- A 7-segment is connected to PORTA. Display 1 on the 7-segment.

DDRC	1	1	1	1	1	1	1	1
PORTC:	0	0	0	0	0	1	1	0

```
    LDI   R20,0b00000110
    OUT   PORTC,R20
    LDI   R20,0b11111111
    OUT   DDRC,R20
L1: JMP L1
```



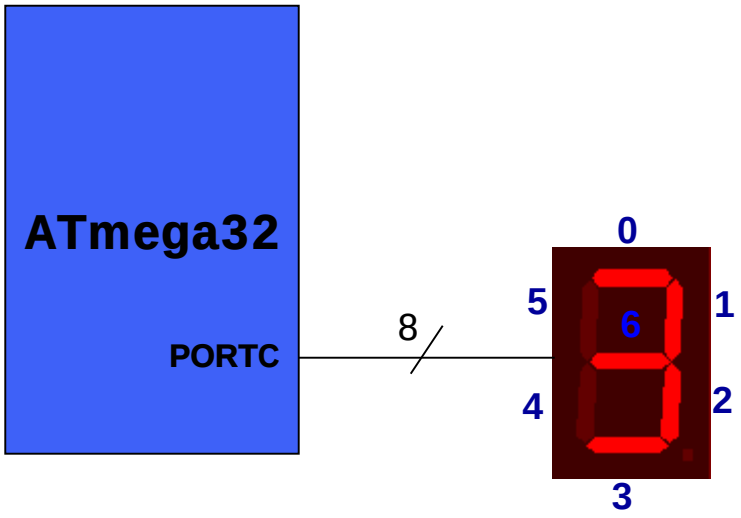
PORTx \ DDRx	0	1
0	high impedance	Out 0
1	pull-up	Out 1

Example 4

- A 7-segment is connected to PORTA. Display 3 on the 7-segment.

DDR:	1	1	1	1	1	1	1	1
PORTC:	0	1	0	0	1	1	1	1

```
    LDI   R20,0b01001111
    OUT   PORTC,R20
    LDI   R20,0b11111111
    OUT   DDRC,R20
L1: JMP L1
```

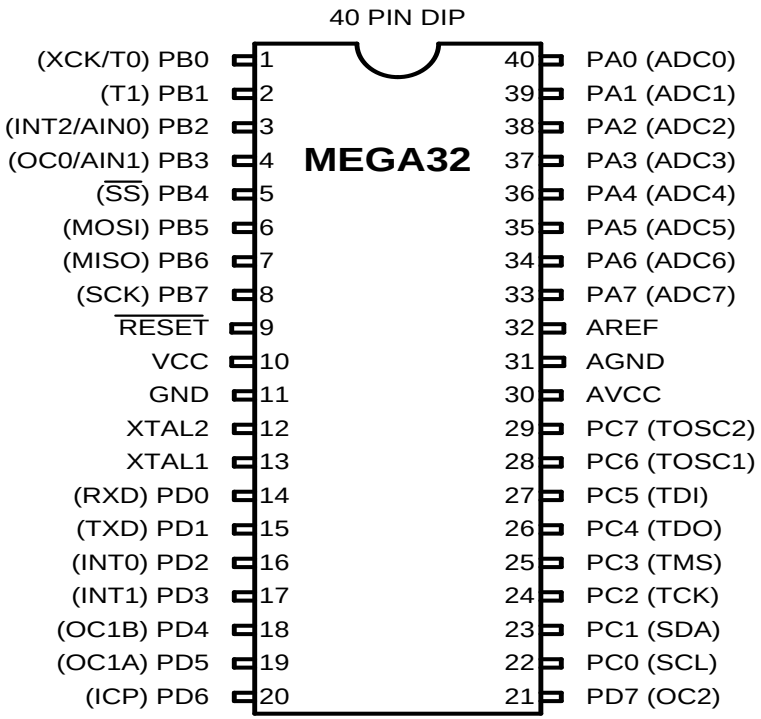


PORTx \ DDRx	0	1
0	high impedance	Out 0
1	pull-up	Out 1

Input eksempel

- Skriv et program, som læser fra PA benene og skriver det til PB benene.

```
     LDI   R20,0
     OUT   DDRA,R20
     LDI   R20,0b11111111
     OUT   DDRB,R20
L1:  IN    R20,PINA
     OUT   PORTB,R20
     JMP L1
```



PORTx \ DDRx	0	1
0	high impedance	Out 0
1	pull-up	Out 1

Single-Bit I/O instruktioner

Table 4-7: Single-Bit (Bit-Oriented) Instructions for AVR

Instruction		Function
SBI	ioReg,bit	Set Bit in I/O register (set the bit: bit = 1)
CBI	ioReg,bit	Clear Bit in I/O register (clear the bit: bit = 0)
SBIC	ioReg,bit	Skip if Bit in I/O register Cleared (skip next instruction if bit = 0)
SBIS	ioReg,bit	Skip if Bit in I/O register Set (skip next instruction if bit = 1)

OBS: Virker kun for I/O registre med adresser lavere end 0x20

SBI og CBI instruktioner

- SBI (Set Bit in IO register)
  - SBI ioReg, bit
  - Eksempler:
    - **SBI PORTD, 0** ;PORTD bit 0 = 1
    - **SBI DDRC, 5** ;DDRC bit 5 = 1
- CBI (Clear Bit in IO register)
  - CBI ioReg, bit
  - Eksempler:
    - **CBI PORTD, 0** ;PORTD bit 0 = 0
    - **CBI DDRC, 5** ;DDRC bit 5 = 0

Eksempel

- Skriv et program, som kontinuert toggler PA ben 4.

```
    SBI  DDRA,4       ;PA ben 4 er udgang
L1: SBI  PORTA,4      ;PA ben 4 = 1 (5 volt)
    CBI  PORTA,4      ;PA ben 4 = 0 (0 volt)
    JMP L1            ;Gentag altid
```

SBIC og SBIS

- SBIC (Skip if Bit in IO register Cleared)
  - SBIC ioReg, bit ; if (ioReg.bit = 0) skip next instruction
  - Eksempel:

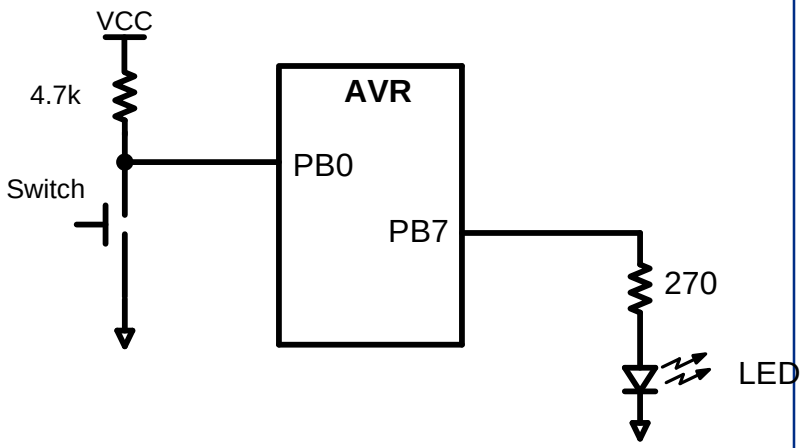
```
SBIC   PIND,0     ;skip next instruction if PIND bit 0 = 0
INC    R20        ;this instruction MIGHT be skipped
LDI    R19,0x23   ;this instruction always executes
```

- SBIS (Skip if Bit in IO register Set)
  - SBIS ioReg, bit ; if (ioReg.bit = 1) skip next instruction
  - Eksempel:

```
SBIS   PIND,0     ;skip next instruction if PIND bit 0 = 1
INC    R20        ;this instruction MIGHT be skipped
LDI    R19,0x23   ;this instruction always executes
```

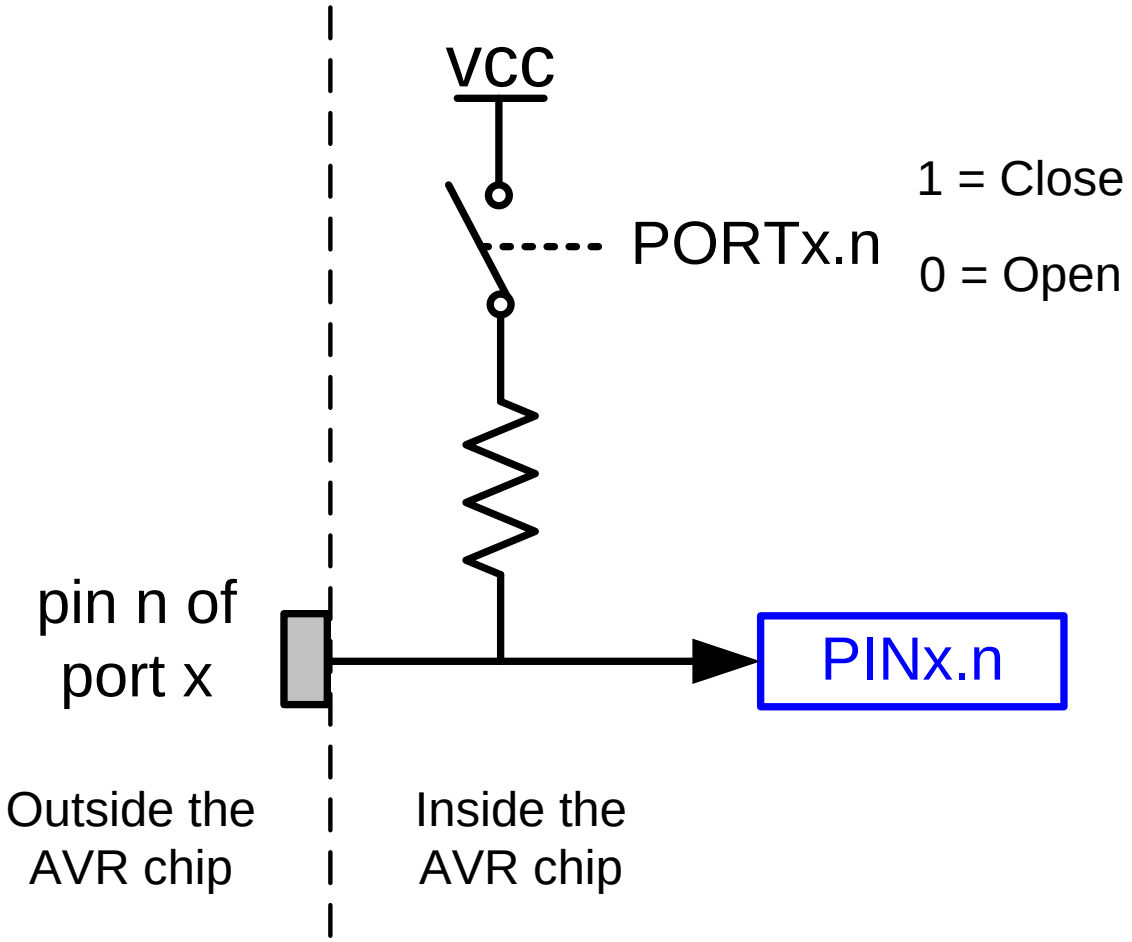
Eksempel

- En switch er forbundet til PB ben 0 og en lysdiode til PB ben 7.
  Skriv et program, der slukker lysdioden, hvis der trykkes på knappen. Ellers skal den lyse.

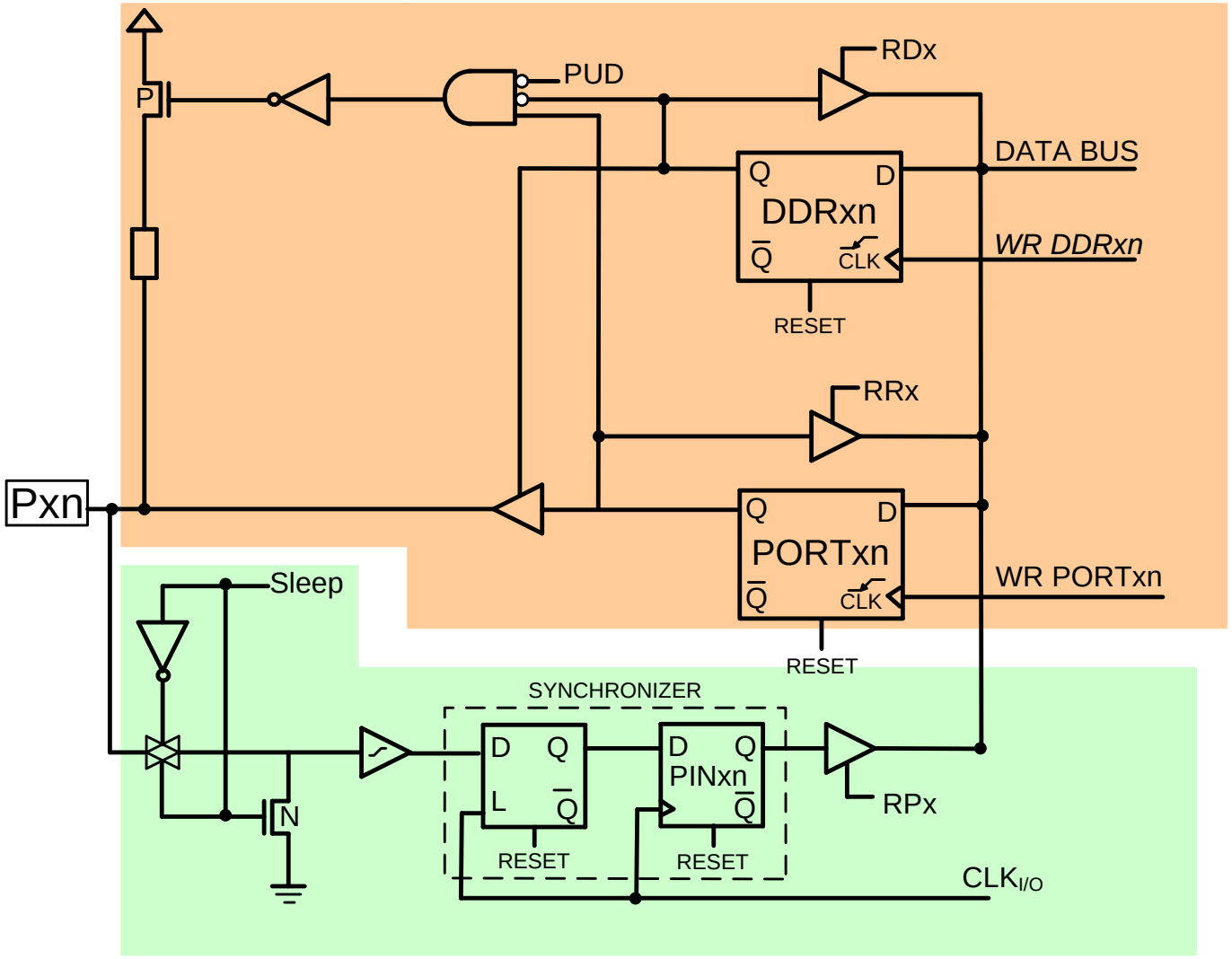



```
        CBI  DDRB,0         ;make PB0 an input
        SBI  DDRB,7         ;make PB7 an output
AGAIN:  SBIC PINB,0         ;skip next if switch PRESSED
        JMP OVER
        CBI  PORTB,7        ;turn off LED
        JMP AGAIN
OVER:   SBI  PORTB,7        ;turn on LED
        JMP AGAIN
```

Pull-up resistor



The structure of I/O pins



Alternative Port-anvendelser (Mega32)

Table 4-3: Port A Alternate Functions

Bit	Function
PA0	ADC0
PA1	ADC1
PA2	ADC2
PA3	ADC3
PA4	ADC4
PA5	ADC5
PA6	ADC6
PA7	ADC7

Table 4-4: Port B Alternate Functions

Bit	Function
PB0	XCK/T0
PB1	T1
PB2	INT2/AIN0
PB3	OC0/AIN1
PB4	SS
PB5	MOSI
PB6	MISO
PB7	SCK

Table 4-5: Port C Alternate Functions

Bit	Function
PC0	SCL
PC1	SDA
PC2	TCK
PC3	TMS
PC4	TDO
PC5	TDI
PC6	TOSC1
PC7	TOSC2

Table 4-6: Port D Alternate Functions

Bit	Function
PD0	PSP0/C1IN+
PD1	PSP1/C1IN-
PD2	PSP2/C2IN+
PD3	PSP3/C2IN-
PD4	PSP4/ECCP1/P1A
PD5	PSP5/P1B
PD6	PSP6/P1C
PD7	PSP7/P1D

CPI - (Compare with Immediate)

Description:

This instruction performs a compare between register Rd and a constant. The register is not changed. All conditional branches can be used after this instruction.

Operation:

(i) Rd - K

	Syntax:	Operands:	Program Counter:
(i)	CPI Rd,K	$16 \le d \le 31, 0 \le K \le 255$	PC ← PC + 1

16-bit Opcode:

0011	KKKK	dddd	KKKK

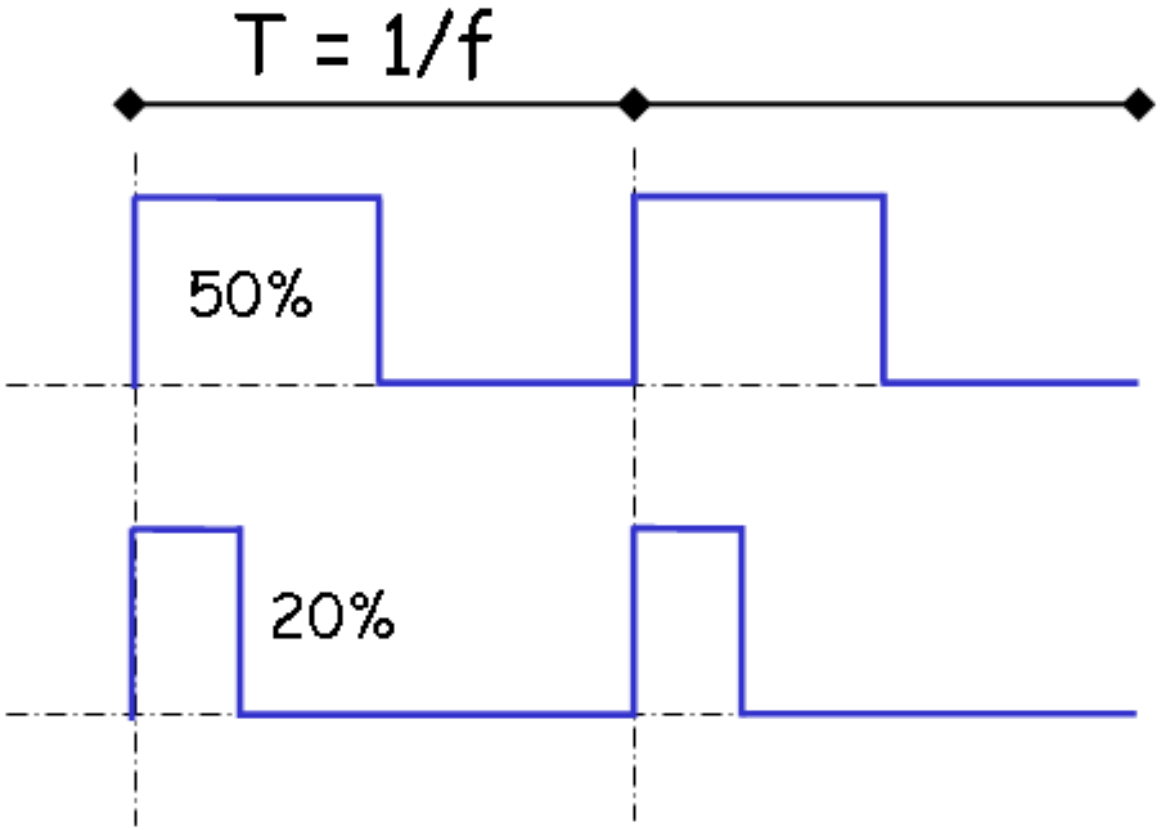
Status Register (SREG) and Boolean Formula:

I	T	H	S	V	N	Z	C
–	–	⇔	⇔	⇔	⇔	⇔	⇔

Example:

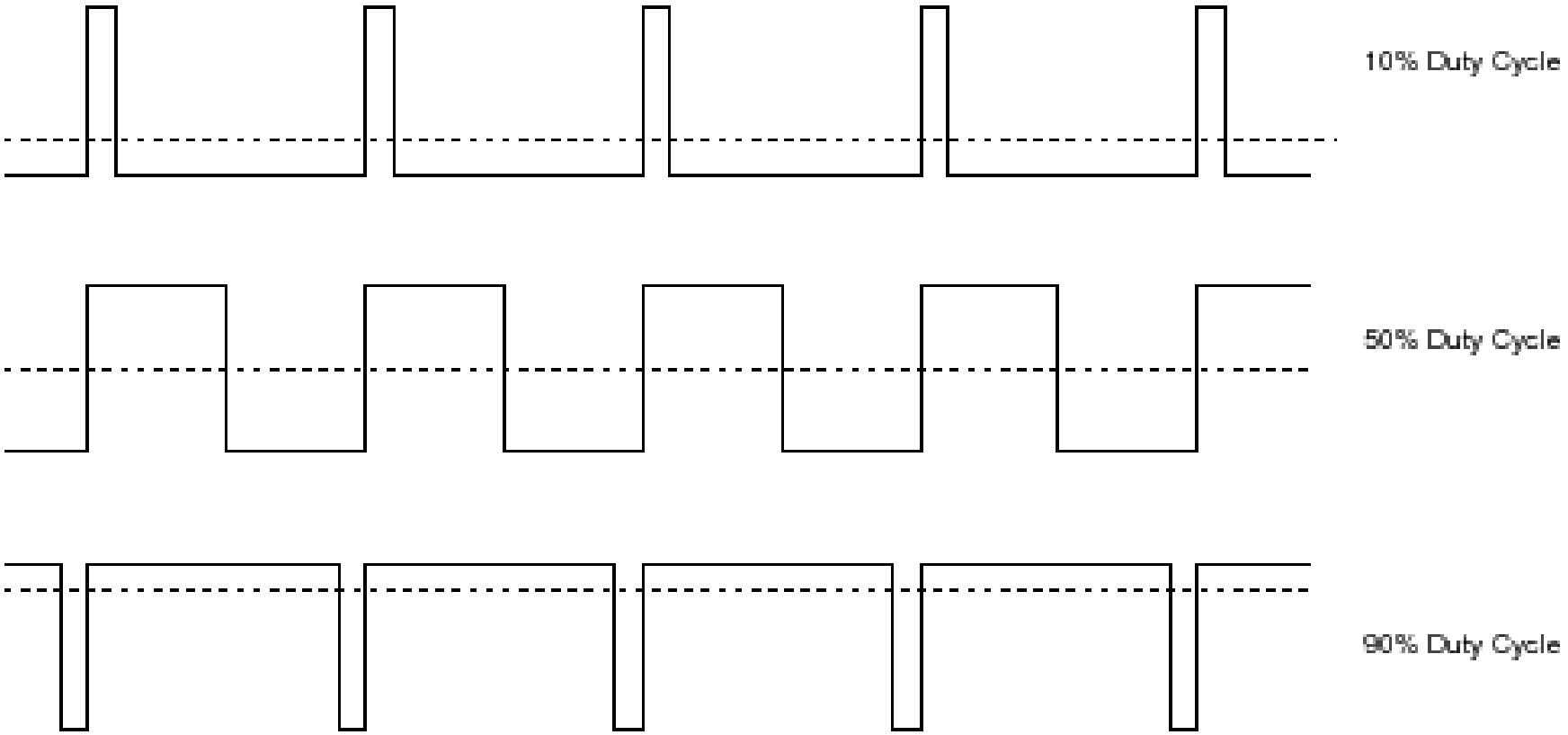
```
            cpi     r19,3     ; Compare r19 with 3
            brne    error     ; Branch if r19<>3
            ...
error:      nop               ; Branch destination (do nothing)
```

PWM = Pulse Width Modulation



PWM: Vi ønsker af styre ”duty cycle”
(frekvensen er mindre vigtig)

Forskellige ”duty cycles”



Ofte er ”average DC” af betydning

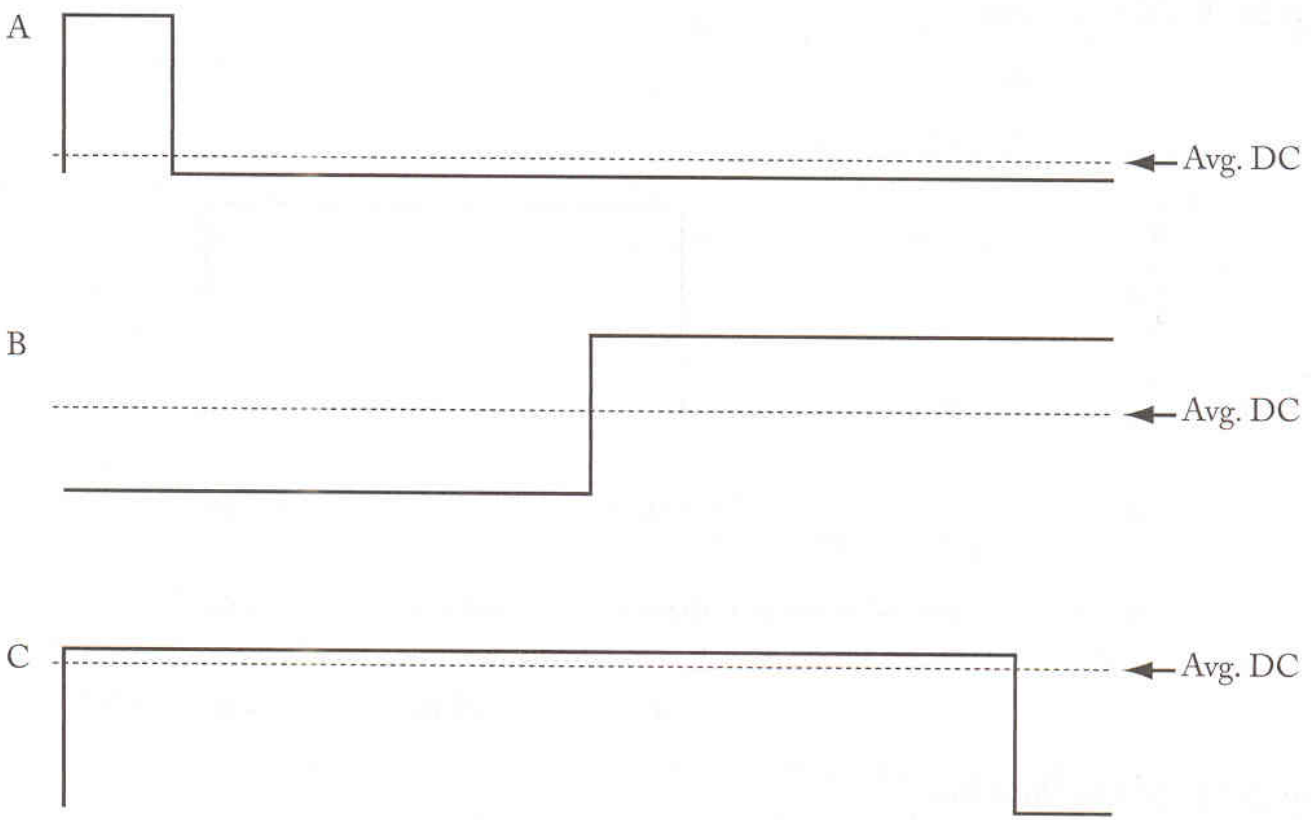
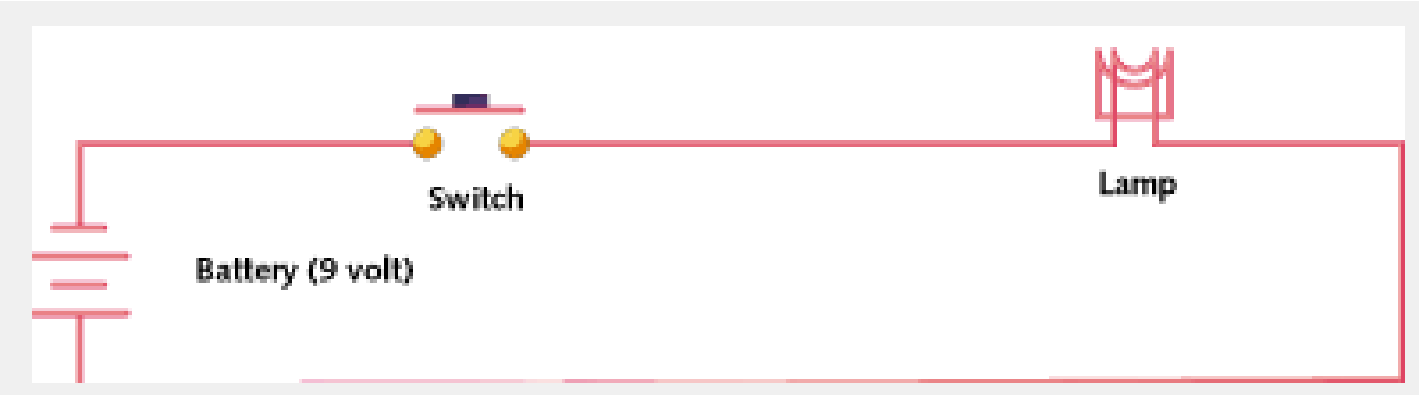
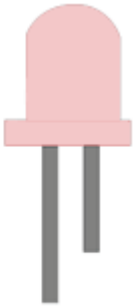


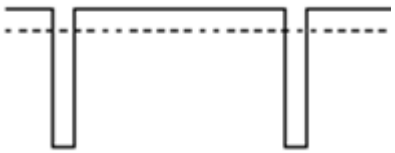
Figure 2–27 *PWM Waveforms*

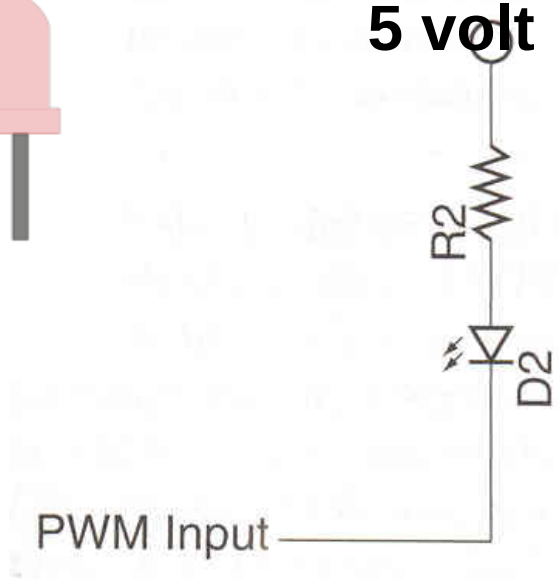
PWM : Styring af lysintensitet



Duty Cycle: 0%







Test (”socrative.com”: Room = MSYS)

- Hvilken duty cycle får vi for signalet på PB bit 6 ?

```
    SBI DDRB,6
IGEN:
    SBI PORTB,6
    CALL DELAY
    CBI PORTB,6
    CALL DELAY
    CALL DELAY
    JMP IGEN
```

A: 33 %
B: 25 %
C: 50 %
D: 66 %

LAB4

Der skal nu laves ændringer i programmet, således at programmet ikke længere ”automatisk” spiller de 8 toner, men vi ønsker i stedet at kunne anvende de 8 trykknapper (”SW7 – SW0”) på ”Mega2560 I/O Shield” som ”tangenter” på et klaver:

SW.: 7 6 5 4 3 2 1 0

Slut på lektion 8