USEFULL INFO

Page 149 List of IO registers (*Table 4-7*)
Page 696 Explanations of assembly instructions

Input Write 00000000b (CLR) to DDRx
Output Write 11111111b (SER) to DDRx

1 instruction takes **271ns** with a clock cycle of **3,6864MHz**

Write **0** to **turn on** a LED
Buttons give **0** when they **are pressed**

1-complement invert all bits (COM)
2-complement invert all bits and add one (COM + INC)

Fuses is used to **change clock frequency** (XTAL) on the timers
ISP In-system programming (*chip can still be programmed when it is in a complete system*)

Watchdog timer resets the system if it is not polled regularly (*resets if the program stalls*)

INTERRUPTS

All interrupts have different priorities defined by their memory locations
SEI enables global interrupts (*C: sei()*)
CLI disables global interrupts (*C: cli()*)

Serial communitation

Baud rate Number of bits sent per second (*bits pr. second*)

Asynchronous
Bits are sent in frames (often 8 bits) with startbit and stopbit(s)
Even parity Number of ones must be even
Odd parity Number of ones must be odd

Simplex One way communication
Half duplex Both way communication, but only one way at a time
Full duplex Both way communication concurrently

UDR Incomming/outgoing data is here

UCSRA Control and Status Register A (*holds flags about transmission*)
UCSRB Control and Status Register B (*control interrupts and enable transmitter/reciever*)

UCSRC Control and Status Register C (*control mode, pairity and character size*)

UBRRH + UBRRL Baud Rate Register (*controls the baud rate*)

Equations for **asynchronous normal mode** (*U2X=0*)

$$Baud\ rate = \frac{f_{oscilator}}{16(UBRR+1)} \qquad UBRR = \frac{f_{oscilator}}{16*[Baud\ rate]} - 1$$

Equations for **asynchronous double speed mode** (*U2X=1*)

$$Baud\ rate = \frac{f_{oscilator}}{8(UBRR+1)} \qquad UBRR = \frac{f_{oscilator}}{8*[Baud\ rate]} - 1$$

Equations for **synchronous master mode**

$$Baud\ rate = \frac{f_{oscilator}}{2(UBRR+1)} \qquad UBRR = \frac{f_{oscilator}}{2*[Baud\ rate]} - 1$$

ADC

ADCH + ADCL Store the result of the analog to digital conversion (*10 bits*)
ADLAR Flag that right or left aligns the result in ADCH + ADCL (*slide 24 lec. 19*)

ADSC Flag der starter konvertering (*ligger i ADCSRA registret*)
ADATE Automatic start (*define mode in SFIOR*)
SFIOR Define the trigger source (*slide 26 lec. 19 || green book p. 51*)

ADCSRA Enable interrupts and holds the interrupt flag

ADMUX For selecting the reference voltage and the input channel
ADCSRA Has the status of ADC and is also used for controlling it
ADCL + ADCH The final result of conversion is here (*10 bits*)

Spændingen (x volt) skal være mellem 0 og referencen der sættes i ADMUX. Hvis spændingen er 0 giver konverteringen 0, og hvis spændingen er lig med referencen giver den 1023 (da resultatet er max 10bit). Alt derimellem er uniformt fordelt.

TIMERS

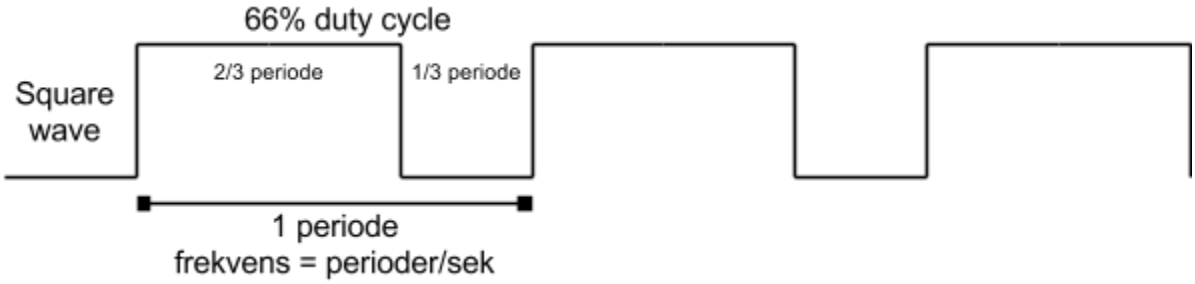
TCNTn counts up with each pulse
TOVn Flag, set when the timer overflows
TCCRn set modes of operation
OCRn used with CTC mode, when TCNTn = OCRn, the OCFn flag is set

TIMSK enable interrupts on overflow or compare match (*timer/counter interrupt mask register*)
TIFR register that holds the flags (*TOVn, OCFn*) (*timer/counter interrupt flag register*)

Normal mode Counts up to max (8bit = 255) and then overflows
CTC mode Counts up to OCR and then resets to 0
PWM mode Counts up and down, changes OCn on compare match with OCRn
PWM fast Counts up and resets/overflows to 0. Changes OCn on compare match and on 0

Delay depends on Crystal frequency (set via. fuses)
The register TCNTn (OCR in CTC mode)
The prescaler (lowest 8, highest 1024)

Duty cycle Hvor stor en procentdel signalet er højt i hver periode
PWM mode used to create non 50% duty cycles



Husk der muligvis skal divideres/ganges med 2 når der snakkes om 50% duty cycle

Mega32 general facts

32 general purpose registers (R0, R1, …, R16, R17, …, R31)
32Kb programmable memory (2k 16bit slots)
1024bytes EEPROM
2Kb SRAM
2 8bit timers (*timer0 and timer2*) and 1 16bit timer (*timer1*)
32 programmable IO lines
40 pins

BRANCHES

SBRC Skip if Bit in Register Cleared
SBRS Skip if Bit in Register Set

BREQ Branch if EQual (Z-bit is 1)
BRNE Branch if NotEqual (Z-bit is 0)
BRLT Branch if Less Than Zero, Signed (N-bit = 1 and V-bit = 1)
BRIE Branch if Interrupt Enabled (I-bit = 1)
BRID Branch if Interrupt Disabled (I-bit = 0)

ASSEMBLY & C

~ = COM
^ = XOR
| = OR
& = AND
>> = LSR (Logical Shift Right)
<< = LSL (Logical Shift Left)